UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING

Electronics and Computer Science

Learning and Evaluation of Topics via Distributional Semantics

by

Alexandry Augustin

A Thesis submitted for the degree of Doctor of Philosophy

December 15, 2020

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
Electronics and Computer Science

A Thesis submitted for the degree of Doctor of Philosophy

LEARNING AND EVALUATION OF TOPICS VIA DISTRIBUTIONAL SEMANTICS

by Alexandry Augustin

Written language is a means of communication. It not only shapes our thoughts, written language also helps us communicate information. As the amount of digital text available keeps growing, it becomes increasingly difficult to locate and keep track of specific information of interest. This observation has fuelled the search for sophisticated representations of written text, and methods for learning meaning. In particular, *topic identification* has grown in importance in recent years as an approach to summarise, organise and understand text. Underpinning modern topic identification methods is the framework of *distributional semantics* which is based on the assumption that meaning is associated with use, and in particular, meaning can be learned by examining the contexts in which words occurs.

Motivated by this, we look in this thesis at the broad field of topic identification in text learned via state-of-the-art distributional semantics models. As such, we provide new answers to the complex question of how meaning is used to derive abstract concepts like topics, and how non-expert humans evaluate such abstract concept generated from artificial processes. In more detail, we address three key problems. We first tackle the problem of evaluating the output of *topic models* (a particular kind of topic identification method) on large text corpora by leveraging non-expert annotators to assess the relevance of topics to a set of documents. Second, we develop a new method to assist in the interpretation of topics by providing additional context. In particular, our solution learns topics as collections of sentences extracted from large corpus of unstructured documents. Finally, we identify and track the topic of text collected over time. In particular, we look at text-based dialogues which often consists of short utterances covering a variety of topics.

Contents

List of Figures

List of Tables

List of Algorithms

Declaration of Authorship

I, Alexandry Augustin , declare that the thesis entitled *Learning and Evaluation of Topics via Distributional Semantics* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- none of this work has been published before submission

Signed: **Alexandry Augustin**

Date: **November 2020**

Acknowledgements

As I bring this chapter of my journey to an end, I reflect on the multitude of lessons learned over the past few years. It was an undertaking of many small steps, with the destination progressively coming sharper into focus. With each step of the way came challenges, hard work, dedication, and serendipity; but there also came opportunities to learn and flourish in ways I could not have possibly anticipated. The journey has undeniably rendered the goal far more rewarding. Yet, I also do reflect on the journey ahead, as every end is the beginning of something new. Still, the future holds no promises; but rather exciting unknowns and surprises waiting to be unveiled.

It took a village for this study to come into existence. I am incredibly grateful for the opportunity I have been given to pursue this work at the University of Southampton. First and foremost, I would like to express my gratitude to my supervisory team for their advices and support throughout. To Professor Nicholas R. Jennings for his valuable comments and guidance that lifted this thesis to a level I never would have reached on my own; but also for giving me the freedom to satisfy my curiosity and explore new directions. To Professor Alex Rogers for his sharp insights and impact-driven outlook on research. To Doctor Jonathon Hare for his expertise and pragmatic approach to problem solving. I am particularly indebted to my sponsor General Electric Aviation, and especially to Doctor Honor Powrie and Doctor Rob Callan, for taking interest in my research, in spite of its scope falling outside of their typical line of business; and for extending valuable suggestions during our meetings. I am also grateful to my mentors and colleagues at Toshiba Research Cambridge Laboratory. To Doctor Alexandros Papangelis for taking a chance on me as a junior researcher; and to Doctor Margarita Kotti for closely following my progress and extending my stay beyond what was original agreed. I thank my examiners Professor David De Roure and Doctor Enrico Gerding for the pertinent questions and compelling suggestions during the viva voce.

I am thrilled to have been part of a strong and collaborative research group. I owe thanks to my collaborators, colleagues and friends who turned my years at Southampton into a unique experience. To Matteo Venanzi, with whom I co-authored my first paper, for going above and beyond the call of duty in a variety of ways to help me succeed. To Pavlos for bringing the topic of natural language processing to my attention; but also for his friendship and camaraderie. Pavlos, I will not hold against you the many basketball matches lost trying to defeat you. While it is tempting to list everyone here, I will spare the reader and single out only a few. To Thanos Panagopoulos, Sabin Roman, Garvin Haslett, Frederik Onthehill, Elliot Salisbury, Eddy Maddalena, Jhim Verame, Edoardo Manino, Zoltán Beck, Mohammad Divband Soorati. I have had many stimulating discussions with all of you, and I will especially miss our tea breaks and lunch time conversations.

Last but not least, I would like to express my love and gratitude to my family for supporting me unconditionally. Friendships come and go, but family is forever.

À Jeanne, Daniel et Justin; pour tout

Chapter 1

Introduction

The information age has altered the ways in which we communicate and placed an increasing emphasis on written digital communications [Geisler et al., 2001]. Written digital communications involves any type of interaction that make use of written words in digital mediums such as ebooks, dialogues, forums, articles, memos, or emails. This evolution in the way we communicate has brought about the challenge of extracting meaningful information from the raw data. While a number of methods are aiming at representing the data in novel ways that are both understandable and useful to humans (e.g. document clustering [Jaiswal and Janwe, 2011], or visualisation [Yang et al., 2008]), the notion of topic plays an important role in summarising, organising and understanding information at both large and small scale [Blei, 2012]. In essence, a topic is a persistent and salient theme across a *semantic unit*, that is, a segment of text of arbitrary size such as a word, an n-gram, a phrase, a sentence, a paragraph, or a document. When used as a summarisation abstraction, topics are captured with a number of different representations depending on the nature of the task. These representations often assume that a tightly connected and easily interpretable set of coherent semantic units act as a hint to identify the topic when presented in an appropriate context. *Topic models* [Blei, 2012] offers one such representation that will be explored in this thesis (Chapter 4). Such models represent topics as a group of words that frequently appear in the same context and are likely to represent a coherent theme. Words such as "Berlin", "Madrid" or "Paris" can clearly be interpreted as the coherent unit "European capital cities" since the words are meaningful enough to understand the main theme. Extractive summaries are another type of representation that this thesis will explore (Chapter 5). In particular, the *multi-topic extractive multi-document summary* (MTEMDS) of a collection of documents is a selected subset of sentences such that the themes of the original documents are preserved in the best possible way [Nenkova and McKeown, 2012]. In addition, topics have also been expressed with a single succinct phrase or a single word (e.g. "international commodity trading" or "justice") to reduce the cognitive overhead associated with interpreting these lists of linguistic units. Such short representation can be derived directly from the more complex representation as lists of semantic units [Mei et al., 2007, Ramage et al., 2009, Magatti et al., 2009, Lau et al., 2011, Bhatia et al., 2016], identified from an external knowledge base of candidates [Tiun et al., 2001, SinghRathore and Roy, 2014], or extracted from within the corpus itself (e.g. keyphrase [Hammouda et al., 2005] or title [Yildirim et al., 2016]). Such topic representations using a documents' title will also be investigated in this thesis (Chapter 6). Since assigning topics manually is a tedious, time-consuming, error-prone, and expensive process that requires knowledge of the subject matter, automatic extraction techniques are of great benefit. For instance, web directories (e.g. Yahoo! Directories or Google Directory) has historically relied on human editors to classify websites into pre-defined categories. However due to the manual labor involved, they have now been supplanted by search engines which automate this process [Chen and Dumais, 2000].

Major advances in automatic topic identification [Blei et al., 2003, Minaee et al., 2020] has led to a number of useful real-world applications over the years. The widespread adoption of content-based recommendation systems, which are used to provide suggestions for book purchases[1] on commercial platforms such as Amazon or Google Books[2], has transformed the way we discover content today. To take one example, recommendation system such as Clavis[3] introduced in 2015 by the Washington Post[4], categorises new stories by topics and extracts a series of keywords which are likely indicators of their content. In turn, these topics are used to suggest follow-up articles for further reading. What's more, with now billions of voice assistants in use today[5] (e.g. Microsoft Cortana, Apple Siri, or Amazon Alexa), human-computer dialogues have emerged from being a niche market (e.g. customer support or flight and hotel booking), to become widely accepted as a more intuitive and engaging way to interact with electronic devices. Until recently, they could only take part in narrow and limited problem domains such as answering simple queries with clear meanings, or carrying out simple commands. Now, open-domain conversational agents engage in meaningful and coherent conversations, and generate responses that are relevant to an unlimited number of topics. Modern content-based recomendation systems and human-computer dialogue systems have in common the use of semantic spaces (Section 1.1) to infer the topics. As such, one of the aim of this thesis is to contribute in further the connection between these two applications. However, despite the clear adoptions of topic identification models, they are subject to shortcomings when used in practice for reasons that will become clear throughout the rest of this chapter.

The remainder of this chapter is organised as follows. We first discuss a number of methods that emerged to identify topics using distributed semantic models in Section 1.1. In Section 1.2, we discuss key aspects related to the evaluation of topic models which

[1] https://patents.google.com/patent/US9646109

[2] https://books.google.co.uk/

[3] http://knightlab.northwestern.edu/2015/06/03/how-the-washington-posts-clavis-tool-helps-to-make-news-personal/

[4] https://www.washingtonpost.com/

[5] https://www.salesforce.com/blog/2019/08/chatbot-statistics.html

lay the foundation of one of the problem we address in this thesis. In Section 1.3, we discuss the aims and challenges of our work. In Section 1.4, we state our contributions and list the respective academic publications that emerged from this work. Finally, in Section 1.5, we outline the structure of the remainder this thesis.

1.1 Distributional Semantics and the Geometry of Meaning

The process by which abstract concepts such as topic is extracted from text feels effortless to the readers. However, this feeling is misleading as this process is not well understood and therefore complex to reproduce artificially [Pexman et al., 2013, Yap et al., 2012, Rabovsky et al., 2012]. Nonetheless, in order to study it, careful computational models capturing meaning must be devised [Recchia and Jones, 2012]. On the one hand, traditional linguistic approaches of *formal semantics* has produced sophisticated mathematical models based on predicate logic that represent the meanings of sentences and how the meaning in different sentences relate to one another. For example, the sentences "everyone who loves someone is happy" and "anyone who is happy is healthy" can equivalently be represented with the expressions $\forall x\,(\exists y\,\text{love}\,(x,y) \rightarrow \text{happy}\,(x))$ and $\forall x\,(\text{happy}\,(x) \rightarrow \text{healthy}\,(x))$. A model then assigns semantic values to each such expression. On the other hand, *distributional semantics* has gained enormous popularity in recent years as a framework to capture meaning directly from data. This popularity is partly explained by their ability to reformulate the problem in standard statistical machine learning terms (e.g. inference, training, prediction, or evaluation) [Blei et al., 2003, Bengio et al., 2003]. Compared to formal semantics, this reformulation makes distributional semantics more desirable for the current paradigm shift in computing from logic-based to data-driven approaches. *Distributional semantic models* (DSMs) often represent semantic units as vectors in a continuous semantic vector space (or *semantic space* for short) [McGregor et al., 2015]. Vectors that are close together in this space are semantically similar (i.e. they mean the same thing), and vectors that are far apart are semantically distant. Sub-spaces corresponding to topics can then be identified based on the geometry of the semantic space, along with their constituent members (i.e. the semantic units). Such a mathematical framework of vector spaces and linear algebra provides a natural framework for talking about distance and similarity between semantic units. It is important to note that, not all vector spaces of semantic units form a semantic space. A defining property of semantic spaces is that the values of their constituent vectors must be induced from statistical regularities in real world corpora. This property is grounded on the *distributional hypothesis* [Harris, 1954] which states that semantic units that occur in the same contexts tend to have the same meanings. Besides this common framework, differences exist in the specific mathematical and computational techniques used to construct such semantic spaces. Recent research has pushed the development of these techniques in two distinct directions [Baroni et al., 2014] that will be looked at in detail in this thesis.

First, *prediction-based DSMs* [Bengio et al., 2003, Mikolov et al., 2013, Le and Mikolov, 2014, Kiros et al., 2015], which will be explored in this thesis (Chapter 5 and 6), predict a given semantic unit from its neighbours resulting in vectors which have been referred to as *embeddings* [Bengio et al., 2003]. They were first developed as an intermediate representation in language modelling within the neural-network community [Bengio et al., 2003]. However over the years, simpler neural network architectures have been developed alleviating the need for more complex architectures to create embeddings that performs equally well [Mikolov et al., 2013, Le and Mikolov, 2014]. Prediction-based models are currently successfully applied in cutting-edge applications enhancing textual representations with extra-linguistic input such as visual [Herzog et al., 2015, Frome et al., 2013, Socher et al., 2013], or auditory information [Kiela and Clark, 2017].

Second, *count-based DSMs* (also called *topic models*) [Deerwester et al., 1990, Blei et al., 2003] compute the frequency of words or n-grams in a corpus resulting in a *word-document matrix* where rows correspond to documents and columns correspond to words. There are a number of approaches for determining sub-spaces corresponding to topic from the word-document matrix. In particular, these methods assume that topics are latent variables that are not directly observed. Factorisation methods such as non-negative matrix factorisation [Kuang et al., 2015], singular value decomposition [Deerwester et al., 1990], or latent variables modeling [Blei et al., 2003, Hofmann, 1999] have successfully been used to uncover topics by decomposing the word-document matrix. From these decompositions, *word spaces* and *document spaces* can be constructed where the topics are the axis, and the words or the documents are the vectors. By doing so, multiple topics are assigned to each word and each document. Beyond their widespread application in text mining, topic models have received a lot of attention in many other research areas such as computer vision [Fei-Fei and Perona, 2005, Luo et al., 2015] or bioinformatics [Liu et al., 2016-12].

Despite their popularity, topic models offer a somewhat impoverished representation. One shortcoming is the fundamental assumption that words are statistically independent of one another. That is, the order of the words is typically ignored. This assumption, referred to as the *bag-of-words* assumption, is clearly unrealistic but avoids computational complexity. Two semantic units may have the same words but be about different topics. For example, “house cat” is a cat kept indoors, while a “cat house” is a shelter intended for a cat. It has been estimated that 80% of the meaning of English text comes from the choice of words, and the remaining 20% comes from their ordering [Landauer, 2002]. Another major shortcoming is that topic models require the specification of the number of topics to discover ahead of time [Greene et al., 2014]. While estimating an appropriate number of topics for a small corpus is reasonable, it becomes unfeasible when the size of the corpus is large. Finding the optimal number of topics for a given corpus has remained an open research question [Zhao et al., 2015, Greene et al., 2014]. As such, topic models may generate topics that are difficult to interpret. In other words, they may not produce semantically coherent concepts that correlate with human judgments [Chang et al., 2009, Mimno et al., 2011]. They may confuse two or more topics into one, have near duplicate topics, generate topics that are not interpretable, and associate words or n-grams with incorrect topics. With the increased popularity of topic models, their evaluation is becoming more important. For this reason, one of the research aims in this thesis is to advance the state-of-the-art in the distributed human evaluation of topic models (Chapter 4).

1.2 Distributed Human Evaluation of Topic Models

The performance of topic models are often assessed using objective metrics borrowed from the fields of statistics and machine learning (e.g. perplexity, held-out likelihood [Blei et al., 2003, Blei and Lafferty, 2007], or other task-specific metrics [Blei et al., 2003]), but rarely relate results to the cognitive assessment of humans [McGregor et al., 2015, Wallach et al., 2009]. It has been shown that traditional objective metrics of topic coherence are negatively correlated with human judgments [Chang et al., 2009, Lee et al., 2017]. For example, low predictive perplexity can result in topics that are overly specific and hard to interpret [Chang et al., 2009]. This has been identified as one of the major obstacle to the wider acceptance of topic models by non-experts outside of the machine learning community [Mimno et al., 2011]. Now, there are a number of human-in-the-loop methods to directly improve on the performance of topic models. One such method enable annotators to interactively provide feedback to the models by, for example, adding or removing words in topics [Hu et al., 2014], splitting generic topics, merging similar topics [Jaegul Choo et al., 2013], or reassigning documents to other topics. Other methods leverage prior human knowledge by using predefined topics [Zhai et al., 2009] or pairs of words that should or should not belong together in a given topic [Andrzejewski et al., 2009]. One promising opportunity that we investigate in this thesis (Chapter 3 and 4) lies in the use of *crowdsourcing* [Howe, 2009], where individuals from the general public brings collective experience to solve a common task. Following the *wisdom of the crowds* principle, groups of individuals are likely to produce better solutions than any single expert by bringing different perspectives to problem solving [Surowiecki, 2005, Page, 2007, Howe, 2009]. However this carries its own set of challenges. Humans have implicit biases influenced by a number of environmental factors such as life experience, culture or upbringing. Furthermore, there are likely to be malicious participants (i.e. spammers) on crowdsourcing platforms who provide judgments randomly. It has been estimated that up to 45% of workers may fall into this category [Vuurens et al., 2011]. As such, methods to reduce the influence of such responses needs to be devised to increase on the accuracy of the aggregation. As experts are less likely to be available, the democratisation of human input becomes imperative.

To date, most research in crowdsourcing has focused on classification in which annotators are asked to provide a single judgment from a finite set of alternatives [Deng et al., 2014]. In the context of topic identification, this translates easily in classifying a semantic unit in a single discrete category (e.g. sport, politics, economics). Several approaches have been proposed to increase the aggregation accuracy and reduce the number of participants within the crowd. Specifically, *plurality voting* represents the simplest consensus method which often performs remarkably well in practice [Sheshadri and Lease, 2013, Tang and Lease, 2011]. This approach simply selects the most frequent judgment provided by the annotators. By so doing, it assumes high quality annotators are in the majority and operate independently which is often not the case in practice. Furthermore, it treats all annotators as equally reliable and does not provide any meaningful measure of confidence in the classification to account for conflicts in judgments or low accuracy. Another strategy known as *weighted majority voting* [Littlestone and K. Warmuth, 1989] introduces weights to capture how good each annotator is. This takes into account the uncertainty over which annotator is the most accurate. Unfortunately, the weights alone are not sufficient to measure the inherent reliability of an annotator. An annotator may be careful but biased, giving consistently and predictably incorrect judgments that can be recovered [Ipeirotis et al., 2010, Welinder et al., 2010].

To overcome this problem, probabilistic methods have been developed which learn the accuracy and bias of each annotator and aggregate their judgments accordingly. These models usually focus on modelling the reliability of individual annotators by way of latent *confusion matrices*, which assign the probability that an annotator provides the correct judgment. In particular, the first proposed solution was based on the expectation maximisation algorithm [Dempster et al., 1977] to jointly learn the accuracy and bias of the annotators while inferring the correct judgments for multiple semantic units at a time [Dawid and Skene, 1979]. Building on this work, the *independent Bayesian classifier combination* (IBCC) [Kim and Ghahramani, 2012] introduced prior probability distributions over the model parameters. This Bayesian extension is helpful when the accuracy and bias of some annotators has been identified a priori enabling the selection of the appropriate prior over the confusion matrices to achieve more accurate classification.

Now, as discussed in Section 1.1, topic models assign multiple topics to each document. As this thesis will make clear, single labels aggregation models such as IBCC are not able to combine the assessment of topic models (Chapter 3). Constraining the annotators to select the most prominent topic among multiple alternatives leads such models to infer inaccurate annotators' ability and aggregation. A limited number of authors have considered providing annotators with the option to select multiple topics per documents in the context of crowdsourcing [Duan et al., 2014, Geng, 2016, Bragg et al., 2013, Deng et al., 2014]. However these approaches either assume a natural hierarchy between topics

[Deng et al., 2014, Bragg et al., 2013], does not account for uncertainty in the value of the models' parameters [Duan et al., 2014], or fail to explicitly address the problem of spammers [Geng, 2016]. Given the importance of spammers in crowdsourcing, we do not adopt these approaches, and instead extend IBCC to handle documents with multiple topics (Chapter 4).

1.3 Research Aims and Challenges

The shortcomings left unaddressed by existing work leads to the following aims and challenge for this research.

Aim 1. Distributed human evaluation of topic relevance. As discussed in Section 1.2, while objective measures are valuable at assessing topic models, they do not always correlate with human judgment of topic quality and relevance [Chang et al., 2009]. Given this, the deviation in the estimation of topic proportions as identified by topic models with judgments from non-expert humans via crowdsourcing needs to be examined and quantified. Now, the aggregation of judgments obtained from open platforms remains a challenging undertaking due to the unknown bias and motivations of the participants. Spammers may intentionally provide incorrect judgments to undermine the aggregation accuracy. Furthermore, crowdsourcing aggregation models providing annotators with the ability to assign multiple topics to a single document has attracted limited attention in the literature. Therefore novel methods must be devised to enable such responses while at the same time account for spammers.

Aim 2. Improving topic interpretability with sentence-based topics.

Interpreting topics represented as lists of words may not always be straightforward, particularly since background knowledge may be required. Therefore, another challenge is to assist in the interpretation of such lists by providing alternative representations richer in context. Sentences are one such representation that capture wider topical context than words alone. However, the ability to accurately group sentences by topics and assigning a score indicating their importance is critical. Furthermore, appropriate datasets with relevant ground-truth are not available [Krishna and Srinivasan, 2018]. For that reason, unsupervised or self-supervised methods must be constructed to address this issue.

Aim 3. Tracking topics over time. Topics are not always static but may change over time. This is particularly true in human-computer dialogues (e.g. Microsoft Cortana, Apple Siri, or Amazon Alexa) which are often composed of a series of sub-dialogues covering different topics [Kim et al., 2014]. In this setting, users expect high quality interactions across a wide range of diverse topics. However, the challenge of identifying topics in such setting is aggravated by the fact that dialogue turns (also referred to as *utterances*) are likely to be short and may consist of only a few words. Additionally, topic identification in dialogues is a dynamic process that must be tracked with careful consideration of context. Dialogues may not contain delimitators indicating topic boundaries. Therefore contents discussed in previous utterances must be taken into account as they are likely to have an important influence on tracking the current topic. Furthermore, in contrast to human-to-human dialogues where dialogues are often composed of multiple sub-dialogues bridging a wide range of topics, human-to-computer dialogue systems are often designed to operate over a limited and static set of predefined topics to improve on performance [Lane et al., 2007]. This limited topic coverage has proven to be a challenge for inexperienced users as they do not necessarily know in advance what topics such systems are able to handle efficiently. Users may attempt to formulate utterances that cannot be handled. For this reason, graceful handling of such cases is crucial to provide robustness to such systems against unexpected user inputs while improving user engagement.

Given this, the topic of each utterance in a dialogue session must be sequentially identified as they unfolds in time to capture context. Furthermore, the topic coverage of current approaches need to be extended while retaining the ability to efficiently enable downstream dialogue systems to generate meaningful reply candidates.

Aim 4. Reducing Generalisation Error via Careful Data Collection and Selection.

A model's capacity to generalise is essential to its success. Generalisation refers to a model's ability to make accurate inference from new and previously unseen data. We outline below a number of generalisation challenges that arise from training learning algorithms.

First, large amount of training data plays a critical role in improving the generalisation of prediction-based DSMs relying on complex and deep architecture [Zhang et al., 2016] (Section 1.1). Often however, large datasets are not easily obtainable due to the high costs of data acquisition. When the training data is limited in size, it is usually hard to get meaningful results for new and unseen documents. On the other hand, human evaluation methods for topic models relying on Bayesian inference (Section 1.2) do not assume large amount of data to ensure generalisation. This is achieved by incorporating background information via prior probability distributions to ensure reliable results with smaller datasets. However, selecting such priors remains an obstacle in some settings [Consonni et al., 2018]. Second, most learning algorithms are prone to poor generalisation when trained on imbalanced datasets [Akbani et al., 2004, Klein et al., 2016]. Finding effective methods to address such imbalance remains an open research problem, for which solutions are often specific to each dataset. Finally, when collecting data from human annotators for the purpose of evaluating topic models, steps must be taken to prevent cognitive biases (e.g. anchoring or memory bias) [Eickhoff, 2018]. Such biases systematically induce errors in judgments and may lead to significant negative effects on the evaluation.

To address these challenges, datasets must be collected and selected with careful consideration for size, balance and bias to minimise generalisation error. In particular, DSMs must be trained with very large amounts of textual data from corpora that closely match a number of well defined topics. Second, prior distributions need to be selected with care to avoid incorrect inferences and poor generalisation of the topic evaluation results. Third, equal proportion of each topic needs to be ensured in each of the dataset via re-balancing methods such as re-sampling (i.e. over-sampling or under-sampling). Forth, evaluation metrics need to be choosen cautiously to avoid misleading or sub-optimal results. Finally, human annotations need to be carefully collected accounting for cognitive biases to strengthen robustness against generalisation error.

1.4 Research Contributions

The aims and challenges noted in the previous section (Section 1.3) lead us to a number of research contributions.

C1. We demonstrate for the first time that when faced with judgments related to documents with multiple topics, current crowdsourcing aggregation models for single topics fail to take this fact into consideration, leading to inaccurate aggregation and inference of the annotators' accuracy and bias (Chapter 3).

C2. We introduce a new Bayesian probabilistic crowdsourcing model to support human assessment of topical relevance (Chapter 4). Our model jointly combines judgments provided by annotators regarding the relevance of documents to a set of topics, while weighting those judgments based on the learned annotators' accuracy and bias. The distinctive characteristic of our model is that it allows each annotator to provide their judgment of proportion in the form of a probability distribution over the topics. Furthermore, our approach is sufficiently flexible to learn the annotators' ability in both unsupervised and semi-supervised settings. In particular, the former approach learns the annotators' ability and the documents' proportion simultaneously, making it especially suitable when the ground truth is not available. On the other hand, if the true proportion is known for some of the documents, performance can be improved on new documents using this limited training data. We empirically demonstrate the effectiveness of our evaluation approach on a real-world dataset provided by crowdsourcing annotators.

C3. We define a novel extractive summarisation model that jointly learns, clusters and scores sentence representations by syntactic and/or semantic similarity in large text corpora on a given topic (Chapter 5). More specifically, we first compute an intermediate representation of each sentence (e.g. the embedding) by predicting its context sentences using a prediction-based DSM. We then performs discrete clustering of each sentence representation in a semantic space. We finally assign a similarity score to each sentence which indicates its importance in the topic. A key aspect of our contribution is the non-reliance on labelled data for training. The proposed model is *self-supervised* avoiding huge manual effort in creating the training sets.

C4. We define a new dialogue topic tracking model which for the first time jointly learns the binary decision boundary of whether an utterance belongs to a set of topics known a priori, and a fuzzier mapping of relatedness under a semantic space for topics not known a priori (Chapter 6). In such a space, we use a similarity measure (e.g. Euclidean distance or cosine similarity) to compute the distance between an utterance and the closest Wikipedia[6] article's embedding that we assume represents a topic. Our architecture offers a simple way to share parameters between the two tasks, forcing our model to perform well on both. Furthermore, our approach is computationally fast enabling real-time inference of topics in dialogues as they unfolds in time.

The research presented in this thesis have been published in the following peered-reviewed publications:

A. Augustin, M. Venanzi, A. Rogers, N.R. Jennings. Bayesian Aggregation of Categorical Distributions with Applications in Crowdsourcing. *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. (Chapter 4).

A. Augustin, A. Papangelis, M. Kotti, P. Vougiouklis, J. Hare, N. Braunschweiler. Open-domain Topic Identification of Out-of-domain Utterances using Wikipedia. *Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS). Workshop on Human in the Loop Dialogue Systems (HLDS), 2020.* (Chapter 6).

1.5 Thesis Outline

The remainder of this thesis is structured as follows:

- In Chapter 2, we review existing approaches to identify topics using DSMs. We then pinpoint their shortcomings and existing solutions to deal with them.

[6] `en.wikipedia.org`

- In Chapter 3, we thoroughly investigate existing crowdsourcing aggregation techniques, and identify IBCC as the most promising point of departure for our work. We present IBCC in detail and demonstrate that when faced with judgments from documents with multiple topics, it fails to take this fact into consideration, leading to inaccurate aggregation and inference of the annotators' accuracy and bias (Contribution 1).

- In Chapter 4, we build upon IBCC to address this shortcoming and allows aggregation over multiple topics (Contribution 2). To this end, we also provide an empirical comparison demonstrating how it performs against a range of state-of-the-art benchmark models. In particular, experimental results show comparable aggregation accuracy when 60% of the annotators are spammers, as other state of the art approaches do when there are no spammers.

- In Chapter 5, we introduce a novel approach to learn sentence-based topics from a corpus of unstructured textual documents (Contribution 3). We show an empirical comparison against a set of benchmarks models using real corpora and present our findings.

- In Chapter 6, we propose a new method for tracking topics in dialogues (Contribution 4). We show experimentally on real-world dialogue data that our approach generates comparable performance when identifying topics known a priori than the benchmarks, but more significantly, it is up to about 30% more accurate when predicting topics not known a priori.

- In Chapter 7, we summarise our results and specify future directions that builds upon the work introduced in this thesis.

Chapter 2

Review of Distributional Semantic Models

Distributional semantic models (DSMs) are now ubiquitous in NLP research today. The reason for this success lies in the usefulness of the feature they provide for a number of tasks such as information retrieval, document classification, question answering, and, as is of concern in this thesis, topic identification. This chapter provides an overview of the literature on DSMs, laying the foundations on which we build our contributions in the subsequent chapters. In doing so, we also describe the main challenges of applying such statistical methods to extract topics from text documents. Although prediction-based DSMs are a more recent development of distributional semantics, we discuss them first in Section 2.1 as we address the evaluation of more traditional count-based DSMs in Chapter 3. In particular, we review a number of word embedding models in Section 2.1.1, and sentence and document embedding models in Section 2.1.2. We then specify how topics can be identified using prediction-based DSMs in Section 2.1.3. Section 2.2 introduces count-based DSMs (also known as topic models) and contrast non-probabilistic with probabilistic approaches. We finally examine a number of limitations, challenges and opportunities that DSMs faces in Section 2.3 which will pave the way for our contributions in later chapters. More specifically, we review methods for tracking topics in dialogues in Section 2.3.3, and we discuss approaches for assessing the relevance of topics identified by count-based DSMs in Section 2.3.1.

2.1 Prediction-based Distributional Semantic Models

Good representation of semantic units (e.g. words or sentences) is a core determinant of the performance of topic identification models. As we will see in Section 2.1.3, such representations serve as features to topic identification models which help them in improving learning performance, increasing generalisation, lowering computational complexity, and decreasing storage complexity. Before discussing topic identification, we first review the major prediction-based DSMs used to compute distributed representations of words in Section 2.1.1, and higher level constructs such as sentences and documents in Section 2.1.2.

2.1.1 Word Embedding Models

Words are the smallest elements of text with a practical meaning. In many NLP tasks they have traditionally been represented as *one-hot* vectors, that is, boolean vectors consisting of zeros with the exception of a single one at the index of the word in the *vocabulary* (i.e. the set of unique words in the corpus). This simple representation leads to orthogonal word vectors which does not capture semantic similarities. For example, given the vocabulary $V = \{\text{blue}, \text{green}\}$, the word 'blue' and 'green' will be represented with the vectors $[1, 0]$, and $[0, 1]$ respectively leading to a dot product of zero. Furthermore, such vectors are usually of very high dimensionality, matching the size of the vocabulary. This has the downside of increasing the number of parameters needed when using those vectors as feature in topic identification models.

More recent developments capture similarities between words by estimating their vectors as a by-product of a supervised learning task, where each vector is learned through gradient descent [Williams and Hinton, 1986] by predicting the contexts in which the word appears [Bengio et al., 2003, Mikolov et al., 2010, 2013]. Since similar words occur in similar contexts [Harris, 1954], such an approaches naturally learn to assign similar words to similar vectors. Word vectors learned using prediction-based models have been referred to as *embeddings* [Bengio et al., 2003]. More formally, the i-th word in a vocabulary V of size $|V|$ is represented by an embedding $\mathbf{w}_i \in \mathbb{R}^n$, that is, a n-dimensional vector of real numbers. The optimal embedding size n is the minimal number of dimensions needed to maximise the accuracy of the prediction [Young et al., 2017]. It is usually in the range 50 to 1000, that is, a much smaller dimension than the typical size of a vocabulary (i.e. $n \ll |V|$). Propelling the recent popularity of word embeddings is their apparent compositionality. That is, adding two word embeddings results in a vector that is a semantic composite of the individual words (e.g. $\text{king} - \text{man} + \text{woman} \approx \text{queen}$). Compositionality is observed when certain assumptions are held, such as the uniform distribution of words in semantic space [Gittens et al., 2017].

Prediction-based DSMs are used to learn word embeddings. They are typically based on probabilistic *language models* [Almeida and Xexeo, 2019] that learn an approximate joint probability distribution $p(\mathbf{w}_1, \cdots, \mathbf{w}_m)$ of observing any sequences of words $\mathbf{w}_1, \cdots, \mathbf{w}_m$. However, to reduce complexity over long sequences, approximations such as n-grams introduce a context window of n words such that

$$p\left(\mathbf{w}_1, \cdots, \mathbf{w}_m\right) \approx \prod_{t=1}^{m} p\left(\mathbf{w}_t | \mathbf{w}_{\text{context}}\right). \tag{2.1}$$

The conditional probability distribution $p\left(\mathbf{w}_t|\mathbf{w}_{\text{context}}\right)$, defined over all the words in the vocabulary V, predicts a target word $\mathbf{w}_t$ given its context $\mathbf{w}_{\text{context}}$. The context $\mathbf{w}_{\text{context}}$ might be a window of previous words $\mathbf{w}_{\text{context}} = \{\mathbf{w}_{t-n}, \cdots, \mathbf{w}_{t-1}\}$, a window of next words $\mathbf{w}_{\text{context}} = \{\mathbf{w}_{t+1}, \cdots, \mathbf{w}_{t+n}\}$, or a combination of both $\mathbf{w}_{\text{context}} = \{\mathbf{w}_{t-n}, \cdots, \mathbf{w}_{t-1}, \mathbf{w}_{t+1}, \cdots, \mathbf{w}_{t+n}\}$. The parameters of the conditional distributions $p\left(\mathbf{w}_t|\mathbf{w}_{\text{context}}\right)$ are commonly set by a neural network such that

$$\begin{aligned} \mathbf{p} &= \text{NeuralNet}\left(\mathbf{w}_{\text{context}}\right) \\ p\left(\mathbf{w}_t|\mathbf{w}_{\text{context}}\right) &= \text{Categorical}\left(\mathbf{w}_t; \mathbf{p}\right). \end{aligned}$$

To ensure that $p\left(\mathbf{w}_t|\mathbf{w}_{\text{context}}\right)$ is a valid probability distribution (i.e. $\sum_i p_i = 1$ and $p_i \geq 0$ for all i), a softmax (.) function is used as the last operation of the neural network such that

$$\text{NeuralNet}\left(\mathbf{w}_{\text{context}}\right) = \text{softmax}\left(\mathbf{f}\left(\mathbf{w}_{\text{context}}\right)\right)$$

for some function $\mathbf{f} \in \mathbb{R}^{|V|}$ defined by the network (i.e. its computational graph).

A number of prediction-based DSMs have been proposed to learn $p\left(\mathbf{w}_t|\mathbf{w}_{\text{context}}\right)$ for all t. In particular, the neural language model (NLM) [Bengio et al., 2003] was the first model to introduce the concept of embeddings. The model has a *projection layer* [Mikolov et al., 2013] mapping each discrete one-hot-encoded word vector to a continuous embedding vector (Figure 2.1a). The projection layer is shared across timesteps t such that a given word contributes changes during training to the same embedding regardless of the timestep at which it appears. The embeddings vectors are then concatenated and fed to hidden layer with a tangent hyperbolic (tanh) activation function connected to the softmax layer. At each time step t, the softmax layer outputs the probability distribution $p\left(\mathbf{w}_t|\mathbf{w}_{\text{context}}\right)$ of the next word $\mathbf{w}_t$ given a sliding context window of the n previous words $\mathbf{w}_{\text{context}} = \{\mathbf{w}_{t-n}, \cdots, \mathbf{w}_{t-1}\}$.

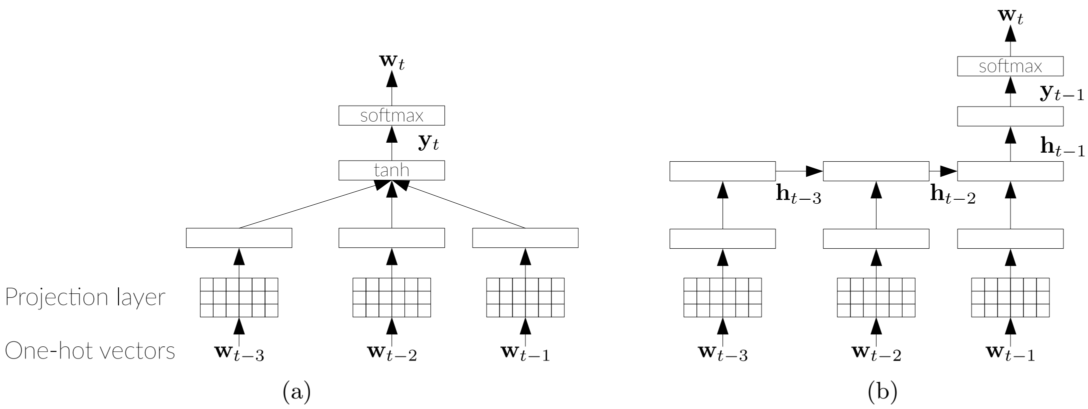
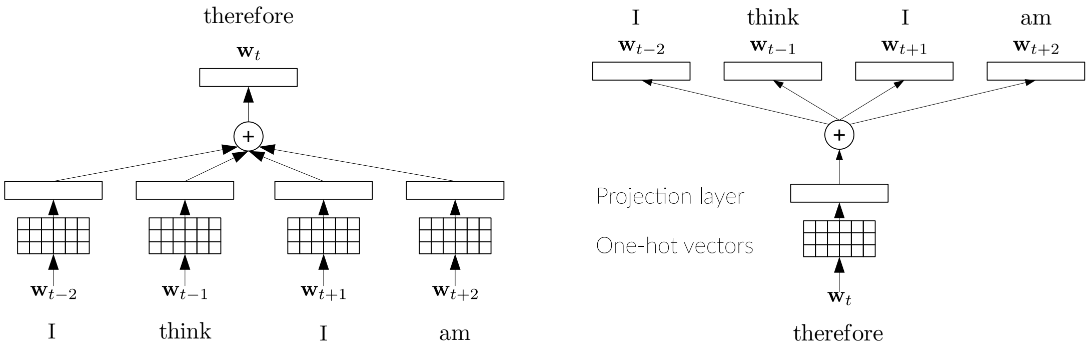


Figure 2.1: Prediction of the next word based on a context window of the previous three words by (a) NLM [Bengio et al., 2003] and (b) RNLM [Mikolov et al., 2010].

More recently, the recurrent neural language model (RNLM) [Mikolov et al., 2010] has been proposed to represent more complex patterns than simpler feedforward networks as used in NLM. Recurrent neural networks (RNNs) [Elman, 1990] (Section 6.1.2) can in principle have an unlimited context size at the cost of higher computational complexity. This advantage enables RNML to significantly outperform NLM on speech recognition tasks [Mikolov et al., 2010], including in scenarios where NLM is trained on larger amounts of data.

Arguably, the immense popularisation of word embeddings was due to the continuous bag-of-words (CBOW) and skip-gram models (collectively known as the *word2vec* model) [Mikolov et al., 2013]. In particular, CBOW is similar to NLM and computes the conditional probability of a target word $\mathbf{w}_t$ given the context words, $\mathbf{w}_{\text{context}}$, surrounding it. On the other hand, the skip-gram model does the exact opposite of the CBOW model, by predicting the surrounding context words, $\mathbf{w}_{\text{context}}$, given the central target word $\mathbf{w}_t$ (Figure 2.2). By removing hidden layers and by approximating the objective function, these architectures provide large improvements in accuracy at a much lower computational cost (i.e. the number of trainable parameters in the model) compared to both NLM and RNLM. While simple, these changes enable training on very large corpora for the first time, which, in turn, improves the quality of the embedding.



(a) CBOW architecture predicts the current word based on the context.

(b) Skip-gram predicts surrounding words given the current word.

Figure 2.2: Word2vec architectures [Mikolov et al., 2013].

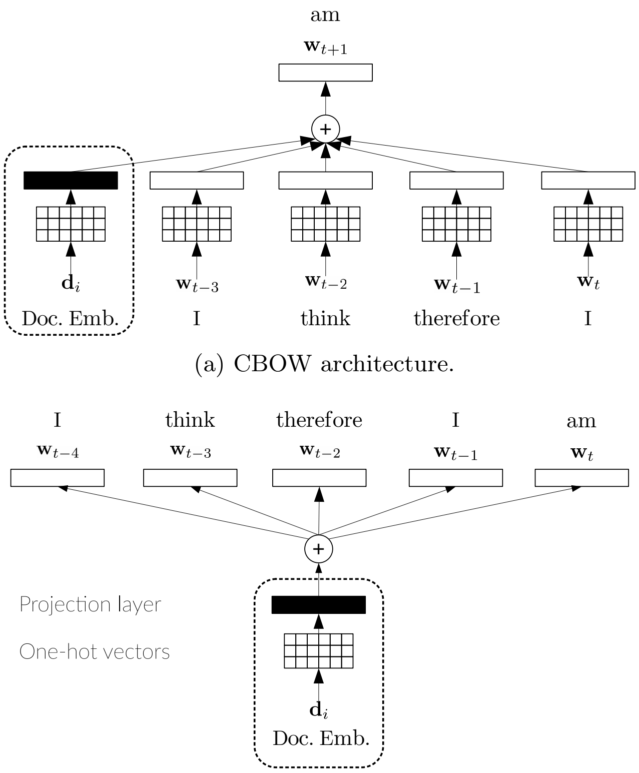
Going a step further, FastText [Bojanowski et al., 2016] extends word2vec by including sub-word features (i.e. character n-grams), which allows more reliable learning of rare words, and words that did not appear in the training corpus. For example, inflected languages (e.g. French or Spanish) remain a challenge to learning good embeddings as words can have many forms. Like FastText, embeddings from language model (ELMo) [Peters et al., 2018] also benefits from sub-word features through the use of character convolutions. In particular, ELMo uses a projection layer at character level connected to a CNN. The CNN representation is then passed through a highway network [Srivastava et al., 2015] (a multi-layer network of arbitrary depth based on a gating mechanism to regulate the information flow across layers without attenuation) before being presented as input to a multi-layer bidirectional LSTMs acting as a language model. The authors observed that lower-level layers capture syntax while higher-level layers capture context. Based on this, the word embeddings are computed as a linear combination of the internal states of the LSTM units across layers. Combining the internal states in this manner allows ELMo to capture rich word representations enabling it to achieve state-of-the-art on a large number of end task. Despite these more recent developments, word2vec is still a popular choice and widely used today.

One limitation of word embeddings is their inability to deal with longer semantic units. Since documents are composed of sequences of words and sentences, it is also important to capture the meaning of longer semantic units for the purpose of identifying topics [Tian et al., 2016, Li et al., 2017].

2.1.2 Sentence and Document Embedding Models

The concept of word embeddings can be extended to sequences of words such as sentences and documents. Similar to word embeddings, sentence and document embeddings are dense vectors that summarise their meaning. There are a number of competing schemes for learning sentence and document embeddings. The simplest approaches compose the embeddings of their constituent words [Blacoe and Lapata, 2012, Mitchell and Lapata, 2008, 2010, Iyyer et al., 2015]. For example, unweighted sum or average achieve good performance in classification tasks [Iyyer et al., 2015] as well as to represent short phrases [Mikolov et al., 2013]. Because this method does not require additional parameters, it is easy to implement. Going a step further, Ruckle et al. [2018] generalise the concept of average word embeddings to the power mean [Hardy et al., 1952] such that the concatenation of the words is given by $\left(\frac{1}{T}\sum_t \mathbf{w}_t^p\right)^{1/p}$ for $p \in \mathbb{R}$. The power mean generalises a number of existing methods, such as the arithmetic mean ($p = 1$), the geometric mean ($p = 0$), and the harmonic mean ($p = -1$). In the extreme cases, the power mean yields the minimum ($p = -\infty$) and maximum ($p = +\infty$) of the sequence. Alternatively, Arora et al. [2017] represent sentences by first computing a weighted average of their word constituent embeddings. They then remove the top n principal components from those sentence representations, and finally project the sentence representation away from the principal components. It is assumed that the top direction inherently encodes the common information across the entire corpus. Therefore, eliminating this component leads to stronger linguistic regularities in the sentence representations (e.g. constant offsets between pairs of embeddings sharing a particular relationship). This method improves performance by about 10% to 30% in sentence similarity tasks [Arora et al., 2017].

While these simple methods give strong results on certain tasks, they fall short of the performances of more complex models. More advanced model such as doc2vec [Le and Mikolov, 2014], draws their inspiration from the word2vec family of models [Mikolov et al., 2013]. Doc2vec treats sentences, paragraphs and documents on equal footing (hereafter referred to as a document), meaning that the length of the word sequence is not taken into account. Training is performed by either predicting the next word given the concatenation of both a context window of word embeddings, and a document embedding (CBOW architecture) (Figure 2.3), or by predicting the context window from the document embedding (skip-gram architecture). While document embeddings are unique to each document, the word embeddings are shared across documents. Inference for new documents is performed by setting the word embeddings, and then training the new document embedding.



(a) CBOW architecture.

(b) The skip-gram architecture.

Figure 2.3: Doc2vec architectures [Le and Mikolov, 2014].

Another successful model to compute sentence embeddings is skip-thought [Kiros et al., 2015]. Skip-thought uses an objective function derived from the skip-gram model [Mikolov et al., 2013] but adapted to sentences. That is, rather than predicting the context words surrounding a target word, the model predicts the surrounding sentences of a target sentence. In particular, the model is based on an encoder-decoder architecture [Cho et al., 2014, Sutskever et al., 2014]. While the encoder maps an input sentence to a vector representation (i.e. it's embedding) (Figure 2.4), two decoders are trained to minimise the error when the previous and next sentences are predicted from this vector.

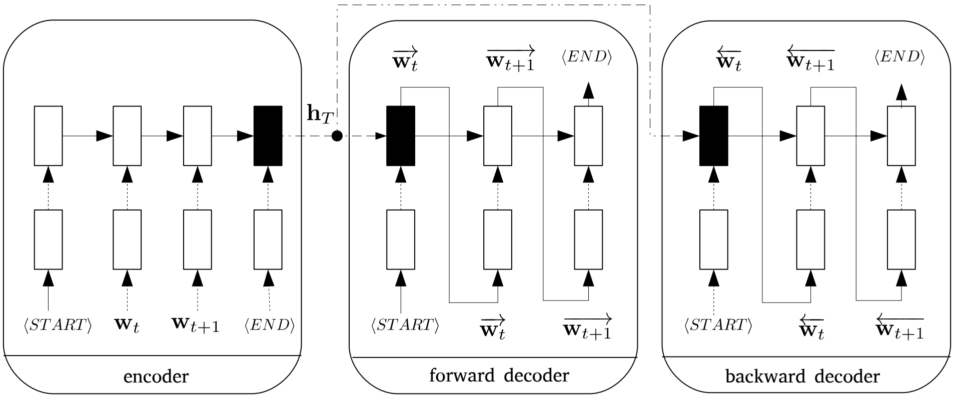

$\mathbf{h}_T$
$\langle START \rangle$
$\mathbf{w}_t$
$\mathbf{w}_{t+1}$
$\langle END \rangle$
encoder
$\overrightarrow{\mathbf{w}_t}$
$\overrightarrow{\mathbf{w}_{t+1}}$
$\langle END \rangle$
$\langle START \rangle$
$\overrightarrow{\mathbf{w}_t}$
$\overrightarrow{\mathbf{w}_{t+1}}$
forward decoder
$\overleftarrow{\mathbf{w}_t}$
$\overleftarrow{\mathbf{w}_{t+1}}$
$\langle END \rangle$
$\langle START \rangle$
$\overleftarrow{\mathbf{w}_t}$
$\overleftarrow{\mathbf{w}_{t+1}}$
backward decoder

Figure 2.4: Skip-thought model [Kiros et al., 2015].

As a result, sentences that have similar syntax and semantics are likely to have similar vectors. During inference, the decoders are discarded and the encoder is used to generate the embedding for new sentences. In particular, the encoder accumulates increasingly richer information as it goes through the sentence, and as it reaches the last word, produces a semantic representation of the whole sentence.

Although skip-thought provides high quality representation of sentences, encoder-decoder based sequence models are known to be slow to train on large corpora compared to shallower networks like doc2vec. To address this, a number of extension to skip-though have been proposed [Tang, 2012, Logeswaran and Lee, 2018, Hill et al., 2016]. In particular, the quick-thought model [Logeswaran and Lee, 2018] replaces the sequential prediction with a classification. The forward decoder is replaced by a classifier which chooses the next sentence among a set of candidates sentences. One strength of this model is its speed of training (an order of magnitude compared to the skip-thought model) making it a competitive solution to exploit larger corpora. On the other hand, models such as FastSent [Hill et al., 2016] exploits the fact that bag-of-words models are more efficiently trained than sequence models by ignoring word order. Given a bag-of-words representation of a target sentence, FastSent simply predicts adjacent sentences also represented as bag-of-words. One downside of skip-thought family of models is their requirement that the training corpus consists of sentences gradually transitioning from one topic to the next. This requirement is problematic to learn from documents with abrupt transitions (e.g. human-agent dialogues or posts from social media). To avoid this restriction, sequential denoising autoencoder (SDAE) [Hill et al., 2016] does away with neighbouring sentences and uses an LSTM-based encoder-decoder architecture to learn a robust representation of a target sentence by encoding a corrupted version following some noise function, and training the model to recover the original version from the corrupted data. The noise function removes words from the target sentence, as well as permuting adjacent words. It can be argued whether SDAE is in fact a DSM since it does not predict the surrounding context of the target sentence. Nonetheless, this approach has shown good performance on paraphrasing tasks compared to other prediction-based DSMs.

Another popular approach learns sentence embeddings via textual entailments. Textual entailment determines whether a relationship holds between two sentences: the text and the hypothesis. The relationship can be: positive (text entails hypothesis), negative (text contradicts hypothesis), none (text does not entail nor contradict). In this regard, InferSent [Conneau et al., 2017] first encodes both sentences into vectors with a shared encoder. Then uses a classifier taking as input the sentence vectors to predict one of the three relations. InferSent demonstrates that leveraging labelled data consistently outperforms self-supervised methods on a wide range of tasks.

As we have seen in this section, a wide variety of architectures for encoding sentences and documents into fixed-size vector representations exists. Table 2.1 gives a more concrete sense of semantic proximity in the space generated by some of the models discussed in this section. As can be seen, it is not always clear which architecture is preferable to best capture meaning. Nonetheless, doc2vec and skip-thought will be used in subsequent chapters to generate sentence and document embeddings for the purpose of identifying topics.

	If he had a weapon, he could maybe take out their last imp, and then beat up Errol and Vanessa.	**An annoying buzz started to ring in my ears, becoming louder and louder as my vision began to swim.**
Sum of CBOW	Then Rob and I would duke it out, and every once in a while, he would actually beat me.	Louder.
Skip-thought	If he could ram them from behind, send them saling over the far side of the levee, he had a chance of stopping them.	A weighty pressure landed on my lungs and my vision blurred at the edges, threatening my consciousness altogether.
FastSent	Isak's close enough to pick off any one of them, maybe all of them, if he had his rifle and a mind to.	The noise grew louder, the quaking increased as the sidewalk beneath my feet began to tremble even more.
SDAE	He'd even killed some of the most dangerous criminals in the galaxy, but none of those men had gotten to him like Vitktis.	I smile because I'm familiar with the knock, pausing to take a deep breath before dashing down the stairs.
Doc2vec	I take a deep breath and open the doors.	They listened as the motorcycle-like roar of an engine got louder and louder then stopped.

Table 2.1: Nearest neighbours of randomly selected target sentences from the Books Corpus dataset [Hill et al., 2016].

2.1.3 Topic Identification from Embeddings

Closer to our work, the main application that embeddings enable is the classification and clustering by topic of semantic units in semantic space. Given a labelled corpus, one can train a classifier to map embeddings to known topics. Alternatively, given an unlabelled corpus, one can use clustering methods to group embeddings together without knowledge of the topics a priori. While classification requires previously defined sets of topics (often manually selected), clustering discovers the groups based only on the raw data without manual intervention. We look at both classification and clustering in turn.

Classification of semantic units by topic has traditionally been achieved via standard classification task using supervised learning approaches. Word features, and in particular bag-of-words features, have been used for this purpose [Baharudin et al., 2010]. However, there have been research efforts to incorporate more complex features into text classification models. In particular, a number of approaches learn embeddings specifically for the task of classification. One such classification method, namely task-oriented word embedding (ToWE) [Liu et al., 2018], learns words embeddings with the objective of generating clearly defined classification boundaries of words in the semantic space. To do so, the model optimises the word embeddings using word2vec taking into account each word importance in the categories. After training, words within the same category are close to each other and far away from words in other categories. Another method, namely the bag-of-embeddings model [Jin et al., 2016], extends the skip-gram model and makes the assumption that the meaning of each word differs across categories. Therefore, rather than learning a single embedding for each word, the bag-of-embeddings model enables multiple embeddings per word according to the category of the document (i.e. multi-prototype embeddings). The probability of a category given a document is then calculated from the probabilities of the embeddings of each word under this category. The model is conceptually simple, with the only parameters being the word embeddings. Another model which captures contextual information while performing classification is Tree-LSTM [Tai et al., 2015]. Tree-LSTM generalises the standard chain structure found in LSTM, to tree structure to classify sentences into categories (Figure 2.5a). While the standard LSTM takes as input the embedding at the current time step and the hidden state of the previous time step, Tree-LSTM takes as input an embedding and the hidden states of arbitrarily many child units. The standard LSTM is then a special case of the Tree-LSTM where each unit has exactly one child. This approach has the advantage of preserving both the word order and the syntactic structure of sentences. However, constructing a tree exhibits additional time complexity, specially when modeling long sentences or a document, making unsuitable in those cases. Deep averaging network (DAN) [Iyyer et al., 2015] is a bag-of-words model (Figure 2.5b) that combine word embeddings with a classification objective. The model first averages the word embeddings, then passes the average through one or more feed-forward layers. Finally the last layer performs a classification with a softmax function. DAN obtains close to state-of-the-art accuracy with much less computational complexity than Tree-LSTM. This is beneficial for capturing semantics of long sentences or documents.

The accuracy of topic classification models plays an important role if they are to be accepted by annotators. However, the output of these models is often limited, consisting only of the classification decision itself. In Chapter 3, we demonstrate the use of crowdsourcing to evaluate topic classification approaches when documents have a single topic.

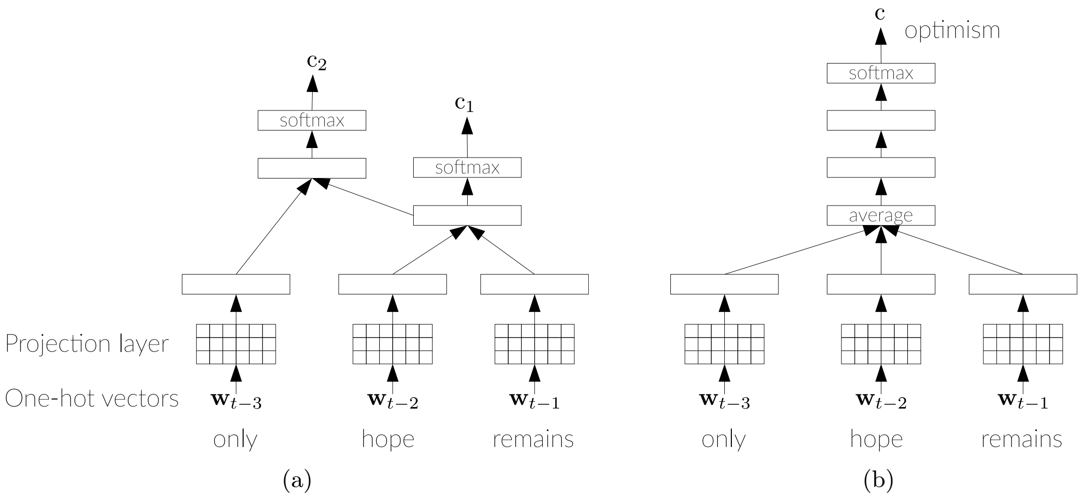


Figure 2.5: (a) Tree-LSTM [Tai et al., 2015] and (b) DAN [Iyyer et al., 2015].

Embeddings have also enabled the organisation of semantic units into clusters. Deep embedded clustering (DEC) [Xie et al., 2016] optimises the clusters and the embeddings jointly. More specifically, DEC clusters the embeddings by simultaneously learning a set of k cluster centers in the semantic space and the parameters of the neural network that maps the embeddings into the semantic space. The method is iterative and assignment to clusters is soft, meaning that embeddings can potentially belong to multiple clusters. Experiments show that the method achieves state-of-the-art clustering results in terms of clustering accuracy and speed. Self-organising maps (SOMs) [Kohonen, 1982] have also been used to group together words [Honkela, 1997, Manukyan et al., 2012] and documents [Kohonen, 1998, Lagus and Kuusisto, 2002, Liu et al., 2012, Kohonen, 2013] (Figure 2.6).

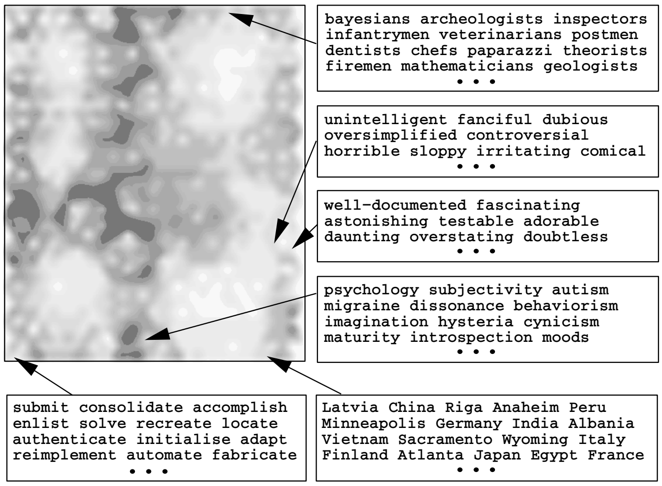

bayesians archeologists inspectors infantrymen veterinarians postmen dentists chefs paparazzi theorists firemen mathematicians geologists • • •
unintelligent fanciful dubious oversimplified controversial horrible sloppy irritating comical • • •
well-documented fascinating astonishing testable adorable daunting overstating doubtless • • •
psychology subjectivity autism migraine dissonance behaviorism imagination hysteria cynicism maturity introspection moods • • •
submit consolidate accomplish enlist solve recreate locate authenticate initialise adapt reimplement automate fabricate • • •
Latvia China Riga Anaheim Peru Minneapolis Germany India Albania Vietnam Sacramento Wyoming Italy Finland Atlanta Japan Egypt France • • •

Figure 2.6: Example of word clustering using a SOM on Usenet newsgroup articles [Honkela, 1997].

Briefly (see Section 5.1.2 for more details), a SOM is a neural network comprised of a one-dimensional input and a two-dimensional output nodes (i.e. weight vectors). During training, each input vector (e.g. embedding or bag-of-words representation [Salton et al., 1975]) is compared to each output node and the winning output node (i.e. the one closest to the input vector) is identified. The winning node and its neighbours are then updated to be closer to the input vector. This process is repeated for a number of iterations, shrinking the size of the neighbourhood window around the winning node at each iteration. As a result, the output nodes self-organise and converge to a topology where neighbouring nodes are more similar than more distant nodes. Being a hard clustering method (i.e. assigning exactly one cluster to each semantic unit), SOMs have the advantage of being fast and easily scalable compared to soft clustering methods such as DEC. Furthermore, they have been shown to be robust to noise and outliers [Lampinen and Oja, 1992]. This means that the addition of a single input to a cluster does not radically change the distances between the output nodes. Although existing models based on SOMs have proven to be useful for the clustering of semantic units, they do not group them by topics explicitly. To address this, Lagus and Kuusisto [2002] take the assumption that each output node encodes the latent topic of the sentences or documents for which the node is a winner. Their approach first encode each sentence or document as a bag-of-words vector [Salton et al., 1975]. The top k winning nodes are then located and used as a semantic representation of the topic. Although this approach clusters sentences and document by topic, it takes as input bag-of-words vectors rather than embeddings. As discussed earlier, embeddings have proven to be a useful representation when capturing meaning which make them a natural choice to identify topics. The use of embeddings as input to SOMs to discover topics will be investigated further in Chapter 5.

Although embeddings are commonly used and have achieved state-of-the-art performances in topic identification and other NLP tasks, the individual embedding dimensions are not interpretable [Levy and Goldberg, 2014]. In fact, the semantic information is distributed across the embedding dimensions making interpretation of each dimension a challenge. There has been little work on gaining a precise theoretical understanding on how embeddings encode meaning. In particular for our work, vector representations that yield interpretable dimensions are desirable if such model are to be used directly by end users for exploratory searches and navigation of large corpora by topics.

2.2 Count-based Distributional Semantic Models

Taking a different perspective to the problem of finding good vector representations of words, count-based DSMs bring light to the semantic information implicitly represented by each vector's dimension. Such models assume that each dimension of a word vector represents a *topic*. In turn, a topic is defined to be a group of words that are likely to appear in the same context and represent a coherent semantic theme such as "genetics" or "data analysis". In more detail, count-based DSMs (also referred to as *topic models*) take as input a corpus of D documents with vocabulary size $|V|$ in the form of a *word-document co-occurrence matrix* $\mathbf{P} \in \mathbb{N}^{|V| \times D}$. The co-occurrence matrix $\mathbf{P}$ is a non-negative matrix where each entry is the count of a particular pair of word and document occurring together in the corpus. This is typically a sparse matrix as many documents contain only a few unique words, or some words appear only in a few documents.

As opposed to prediction-based DSMs, count-based DSMs make the simplifying assumption that words are statistically independent of one another. That is, the order of the words in a document is ignored. This assumption, referred to as the *bag-of-words* assumption, is clearly unrealistic but avoids difficulties in estimating the probabilities of word dependence. A fundamental problem that topic models attempt to solve is to represent the co-occurrence matrix $\mathbf{P}$ in a more compact and meaningful way. In particular, topic models assume there is a latent semantic structure (i.e. topics) in the corpus that can be derived by factorising the co-occurrence matrix $\mathbf{P}$. While the co-occurrence matrix $\mathbf{P}$ is sparse, the corresponding low-dimensional latent factors will typically be dense. This implies that it is possible to compute a meaningful association between words and documents, even if the documents do not have any words in common. The hope is that words with common meaning are mapped to the same region in the latent space. As such, factorisation methods like *non-negative matrix factorisation* (NMF) [Paatero and Tapper, 1994] have been successfully applied to topic modeling [Shi et al., 2018, MacMillan and Wilson, 2017, Kuang et al., 2015]. Given a co-occurrence matrix $\mathbf{P}$, NMF performs the factorisation

$$\mathbf{P} = \boldsymbol{\beta}\Theta^T \tag{2.2}$$

where the matrix $\boldsymbol{\beta} \in \mathbb{N}^{|V| \times K}$ defines the topics, and the matrix $\boldsymbol{\theta} \in \mathbb{N}^{D \times K}$ the topic mixture per document (Figure 2.7).

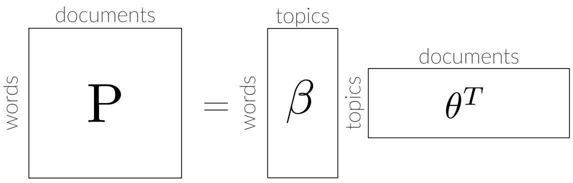


Figure 2.7: Topic Modeling via NMF.

Alternatively, latent semantic analysis (LSA) [Deerwester et al., 1990] applies a *singular value decomposition* (SVD) to factorise the co-occurrence matrix $\mathbf{P}$ such that

$$\mathbf{P} = \boldsymbol{\beta}\Sigma\theta^T. \tag{2.3}$$

In particular, the co-occurrence matrix $\mathbf{P}$ is decomposed into three factors $\boldsymbol{\beta} \in \mathbb{R}^{|V| \times K}$, $\boldsymbol{\Sigma} \in \mathbb{R}_+^{K \times K}$, and $\theta \in \mathbb{R}^{D \times K}$ where β and θ are orthogonal matrices, and $\boldsymbol{\Sigma}$ an ordered and non-negative diagonal matrix (Figure 2.8).

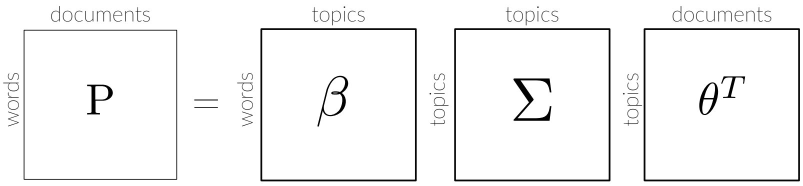


Figure 2.8: SVD factorisation of the co-occurrence matrix performed by LSA [Deerwester et al., 1990].

Now, although NMF and SVD are widely used and well-founded methods for dimensionality reduction stemming from linear algebra, they do not provide the flexibility and advantages of a probabilistic framework. The probabilistic approach takes advantage of the well-established statistical theory for model fitting, model selection and complexity reduction. Consequently, *probabilistic topic models* formulate word and topic vectors as probability distributions. Models such as probabilistic NMF (pNMF) [Luo et al., 2017] and probabilistic LSA (pLSA) [Hofmann, 1999] have been proposed as probabilistic formulation of NMF and LSA respectively. In pLSA, the normalised co-occurrence matrix **P** is the joint probability distribution of words and document $p(w,d)$. By introducing a latent topic variable z for each word, such that given a topic, the words and document are independent (i.e. $(w \perp d)\,|z)$, the pLSA decomposition can then be written as

$$\begin{aligned} p(w,d) &= \sum_z p(w,d,z) \\ &= \sum_z p(w,d|z)\,p(z) \\ &= \sum_z p(w|z)\,p(z)\,p(d|z)\,. \end{aligned} \tag{2.4}$$

To stress the relation with LSA, we can rewrite Equation 2.4 in matrix notation. Letting $\boldsymbol{\beta}_{ik} = p(w_i|z_k)$, $\Sigma_k = \operatorname{diag}(p(z_k))$, and $\Theta_{jk} = p(z_k|d_j)$ gives the LSA decomposition in Equation 2.3. Furthermore, observing that $p(z)\,p(d|z) = p(d)\,p(z|d)$, an alternative decomposition is

$$p(w,d) = p(d)\sum_z p(w|z)\,p(z|d)\,. \tag{2.5}$$

pNMF makes the assumption that the normalised co-occurrence matrix **P** is the conditional probability distribution of words and document $p(w|d)$. From equation 2.5 it follows that

$$p(w|d) = \sum_z p(w|z)\,p(z|d)$$

which is equivalent to Equation 2.2 in matrix notation. Despite this direct correspondence between non-probabilistic and probabilistic topic models, there is a fundamental difference in the choice of objective function used to determine the optimal decomposition. NMF and LSA use the L_2-norm which implicitly assumes additive Gaussian noise on the counts values. In contrast, pNMF and pLSA relies on the maximum likelihood principle and aims at optimising predictions [Hofmann, 1999].

Although pLSA is an important step forward in probabilistic topic modelling, its generative process is incomplete leading to overfitting and problems in assigning probability to unobserved documents [Blei et al., 2003]. In this regard, the latent Dirichlet allocation (LDA) [Blei et al., 2003] solves this problem by introducing a Dirichlet prior over the topic mixture for each document, thus extending pLSA to the full Bayesian paradigm. Being a conjugate prior of the multinomial distribution (Appendix A), the Dirichlet distribution is a convenient choice for reducing computational complexity. In practice, this results in a better disambiguation of words and a more precise assignment of documents to topics. Because LDA is a fully generative model, the dependencies among the variables can consisely be captured by a factor graph [Minka and Winn, 2009] (Figure 2.9).

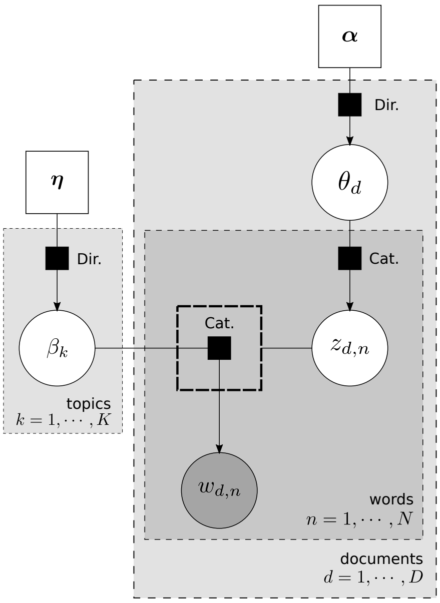


Figure 2.9: Factor graph of LDA.

From the factor graph, the joint distribution can readily be obtained

$$\begin{aligned} P\left(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\beta}; \boldsymbol{\alpha}, \boldsymbol{\eta}\right) &= p\left(\boldsymbol{\beta}; \boldsymbol{\eta}\right) p\left(\boldsymbol{\theta}; \boldsymbol{\alpha}\right) p\left(\mathbf{z}|\boldsymbol{\theta}\right) p\left(\mathbf{w}|\mathbf{z}, \boldsymbol{\beta}\right) \\ &= \prod_{k=1}^{K} p\left(\beta_k; \boldsymbol{\eta}\right) \prod_{d=1}^{D} \left(p\left(\theta_d; \boldsymbol{\alpha}\right) \prod_{n=1}^{N} p\left(z_{d,n}|\theta_d\right) p\left(w_{d,n}|\beta_{z_{d,n}}\right) \right) \end{aligned}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ are parameters of the Dirichlet prior on the per-document topic distributions and the per-topic word distribution respectively. We can illustrate the generative process with a concrete example based on a corpus of scientific articles (Figure 2.10). The words in each document are assumed to be obtained by repeatedly choosing a topic z (coloured circles) from the topic proportions $p\left(z|\theta\right)$ (coloured histogram), and then sampling a word from the corresponding topic $p\left(w|z\right)$ (coloured sheets). All the documents in the corpus share the same set of topics, but each document exhibits those topics in different proportions. Therefore, if the next document in the collection is about "evolutionary biology" and "neuroscience", the model will place a high probability on those two topics.

Because LDA is a fully generative and Bayesian model, it has received considerable research interest and has been extended in many ways since its introduction [Blei, 2012]. One active area of research is to relax and extend its assumptions (e.g. bag-of-word, document and topic independence, or the fixed vocabulary size and number of topics) to uncover more sophisticated structure in the texts. In particular, the correlated topic model (CTM) [Blei and Lafferty, 2007] relaxes the topic independence assumption by capturing topic correlation via a logistic normal prior distribution. The covariance matrix parametrising the logistic normal distribution is used to identify correlations between topics at the cost of higher computational complexity compared to the simpler LDA. As

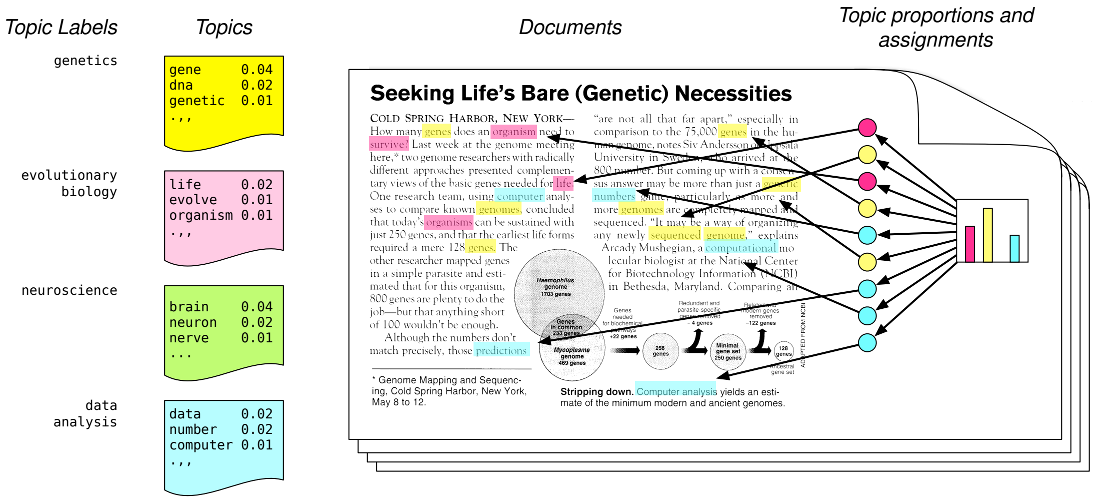

Topic Labels
Topics
Documents
Topic proportions and assignments
genetics
gene 0.04
dna 0.02
genetic 0.01
.,,
evolutionary biology
life 0.02
evolve 0.01
organism 0.01
.,,
neuroscience
brain 0.04
neuron 0.02
nerve 0.01
...
data analysis
data 0.02
number 0.02
computer 0.01
.,,
Seeking Life's Bare (Genetic) Necessities
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
256 genes
Minimal gene set 250 genes
128 genes
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Figure 2.10: Illustrative example of topics and topic assignments in a scientific article about the use of data analysis to determine the minimum number of genes needed for a simple organism to survive [Pennisi, 1996, Blei, 2012].

a result, CTM produces topics that are easier to interpret by humans than those produced by LDA [Chang et al., 2009]. Gaussian LDA [Das et al., 2015] relaxes the fixed vocabulary size assumption in LDA by defining topics as Gaussian distributions in the semantic space formed by word embeddings. Rather than considering documents as sets of one-hot encoded word vectors, as in LDA, Gaussian LDA consider documents as a collection of embeddings computed from prediction-based DSMs. This allows the handling of out-of-vocabulary words since adding new words does not increase the dimension of the embeddings. Gaussian LDA has also been shown to achieve higher topic semantic coherence, that is, it generates topics with words that are more relevant. Neural topic model (NTM) [Cao et al., 2015] alleviate the problem of setting the precise form of the probabilistic dependencies and prior probabilities. The model uses a neural network to model the decomposition of the co-occurrence matrix **P** in Equation 2.2 into simple feed-forward layers.

The topic models discussed so far faces shortcomings when evaluating the interpretability and relevance of the topics they identify. We will discuss these shortcomings in more detail in Section 2.3.1, and propose an evaluation framework in Chapter 4.

Beyond identifying topics, count-based DSMs have also been used to learn word representation akin to embeddings (Section 2.1.1) by computing a *word-to-word co-occurence matrix* [Lund and Burgess, 1996, Rohde et al., 2006, Dhillon et al., 2011, Lebret and Collobert, 2014, Pennington et al., 2014] instead of the word-document co-occurrence matrix. Two of the most influential count-based DSMs for word embeddings are HAL [Lund and Burgess, 1996] and GloVe [Pennington et al., 2014]. On the one hand, HAL computes the word-to-word co-occurence matrix by sliding a fixed-sized context window across the corpus. Words within this window are counted as co-occurring with a strength inversely proportional to the distance from the context word to the target word. On the other hand, GloVe uses the ratios of co-occurrences, rather than the raw counts used in HAL, to encode semantic information about pair of words. Compared to the raw counts, the ratio is better able to discriminate relevant words from less relevant words given a topic.

Despite the success of embeddings computed from prediction-based DSMs, little is understood about the difference with their count-based counterparts. Interestingly however, it has been shown that count-based word embeddings can be linearly projected to the semantic space spanned by prediction-based word embeddings [Bollegala et al., 2017]. This suggest that count-based embeddings contain at least the same information as in the prediction-based embeddings.

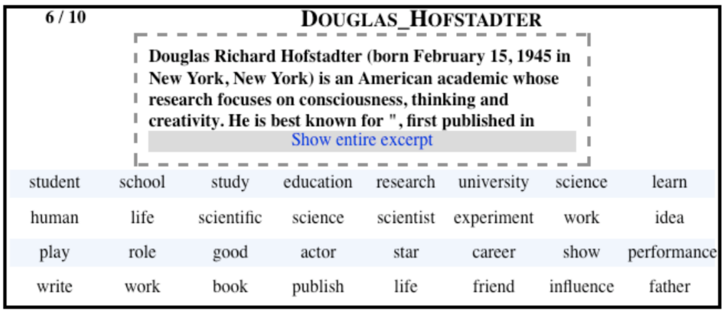
In the next section we examine a number of limitations, challenges and opportunities faced by both prediction-based and count-based DSMs.

2.3 Challenges, Limitations and Opportunities

Although DSMs offer a number of advantages in representing semantic units over bag-of-words or one-hot encodings, they still have a number of issues when used in practice. In this section, we will be looking at some of these issues in turn, and discuss opportunities for improvements.

2.3.1 Topic Relevance

The unsupervised nature of topic models, and therefore the lack of ground truth to compare against, means that the evaluation of the relevance of the topics inferred to the documents remains challenging. While the most common means of evaluating topic models involve measuring the performance on secondary tasks (e.g. document classification or information retrieval), or estimating the probability of unseen held-out documents, a limited number of methods have been proposed for evaluating how humans interpret relevance. On such method, namely *topic intrusion* [Chang et al., 2009] evaluates how well the decomposition of a document into topics agrees with human judgments. Annotators are given a document which consists of the title and the first few sentences (Figure 2.11). Along with the document, they are presented with four topics (each topic is represented by the eight top words within that topic). Three of those topics are the highest probability topics assigned to that document by three models (i.e. pLSA, LDA and CTM). The remaining topic is chosen randomly from the low-probability topics. The annotator is instructed to choose the topic which does not belong with the document. As each of the three models explicitly assigns a topic proportion to each document, this task determines whether humans make the same association.



6 / 10

DOUGLAS_HOFSTADTER

Douglas Richard Hofstadter (born February 15, 1945 in New York, New York) is an American academic whose research focuses on consciousness, thinking and creativity. He is best known for " , first published in

Show entire excerpt

student	school	study	education	research	university	science	learn
human	life	scientific	science	scientist	experiment	work	idea
play	role	good	actor	star	career	show	performance
write	work	book	publish	life	friend	influence	father

Figure 2.11: Topic intrusion task presented to annotators on Amazon Mechanical Turk [Chang et al., 2009].

From this experiment, the authors came to the conclusion that documents about very specific and unambiguous concepts have high agreement between the annotators and the models because it is easy for both humans and the model to assign the document to a particular topic. However, when documents express multiple disparate topics, human judgments diverge from those of the models. At worst, for documents which touch on diverse areas simultaneously, it is difficult for the models to determine specific themes which match human perceptions. The result of this study provides the motivation for our contribution in Chapter 4. Going a step further however, we make the assumption that the subjective analysis of the topic intrusion task leads to different outcomes depending on which individual is doing the analysis. Such diversity needs to be taken into account when results are aggregated.

2.3.2 Topic Interpretability

Unigram topic models are formulated based on the bag-of-words assumption, which leads to simpler and computationally more efficient inference algorithms. This assumption is reasonable for identifying topics, but it becomes a handicap when interpreting them which hampers their adoption. Topic models provide no guarantee that the topics inferred will be interpreted correctly by humans. It has been shown that models with a small number of topics result in broad coverage that can be a mixture of two or more topics. On the other hand, models with too many topics have a very narrow coverage that can often be uninterpretable to non-domain experts [Steyvers and Griffiths, 2007]. Topic interpretability is particularly exacerbated when faced with short documents (e.g. dialogue scripts or messages from social networks) as they require large amounts of data in order to obtain accurate and useful representations. To illustrate, consider the topic extracted from the Reuters' corpus of news articles (Table 2.2). In this example, the list of unigrams (i.e. single words) is insufficiently informative to identify the topic of coffee trading. One could interpret this topic as "agreement in quality of coffee products across regions" or "production of quality coffee products in western countries". Indeed, it is generally challenging to meaningfully convey a coherent and unified theme solely based on a list of unigrams. Such inference requires major feats of interpretation and often lead to ambiguous understanding of the topic due to lack of contextual information [Peng et al., 2016, Huang, 2018]. Unigrams are often part of broader sentences, which are lost in a simple unigram representation. The problem becomes more serious when the user is unfamiliar with the corpus.

Unigrams	coffee, agreement, western, quality, areas, product
Bigrams	coffee export, market price, producer group, consuming countries, export quotas, status quo
Phrases	International Coffee Organization, coffee, largest coffee producers and consumers, improve quality, overseas coffee sales, agreed in principle
Sentences	Talks on the possibility of reintroducing global coffee export quotas have been extended [...]. Brazil is the world's largest coffee producer and exporter. Retail coffee prices over the past year have remained about steady. [...] 173 commercial coffee growers under his association had increased production [...]. The demand was good and all quality coffees were sold [...]. Mexico has temporarily suspended overseas coffee sales due to falling prices [...].

Table 2.2: Illustrative example of the top six unigrams, bigrams, phrases and sentences of a topic about coffee trade (Reuters-21578 dataset).

To address this, a number of approaches extends traditional unigram-based topic models (e.g. LDA or pLSA) to capture more context using n-grams (i.e. sequences of exactly n words) [Wang et al., 2007, Lindsey et al., 2012, Nokel and Loukachevitch, 2015]. While incorporating n-grams in topic models is appealing, such approach often suffer from higher model complexity and data sparsity since the vast majority of n-grams are very uncommon. One of the major drawbacks with n-grams however, is that they do not deal with long distance dependencies. For example, assume that the sequence "coffee quotas" is a frequent bigram (i.e. sequence of two words) in a training set. If the sequence "coffee export quotas" is found in the test set, the bigram is never captured and its associated probability is 0. Although the two sequences are semantically highly similar, an n-gram model does not allow gaps and would not capture the relationship between the sequences. Smoothing techniques (e.g. Laplace or Jelinek-Mercer smoothing) alleviate this issue to some extent, but still requires the use of an arbitrary interpolation of probabilities from observable sequences to non observable sequences. The fixed context-length nature of n-gram models limits their ability to capture longer-term dependencies, therefore preventing longer and more expressive phrases.

Some other methods directly post-process the output of unigram-based topic models to generate phrase-based topics (i.e. collection of short variable-length n-gram). For example, methods such as turbo topics [Blei and Lafferty, 2009] and keyphrase extraction and ranking by topic (KERT) [Danilevsky et al., 2014] first fit a unigram-based topic model, and then annotate each word in the corpus with its most probable topic. A statistical co-occurrence analysis (i.e. nested permutation tests for the former, and FP-growth [Pei, 2004] for the latter) is then carried out to extract the most significant phrases for each topic. Although these approaches preserve the simplicity of bag-of-words models while incorporating features of more complex models, they do not fully fulfill our requirement (Aim 2) as sentences are a richer form to express precise semantic than phrases. Referring back to our example in Table 2.2, by using sentences it is clear that the unigram "quality" refers to "coffee" and not "agreement", and further that the overall topic refers to the trading of coffee.

Beyond post-processing the output of unigram-based topic models, current literature on topic models does not address the problem of identifying sentence-level topics directly. For this, we turn to the field of multi-topic extractive multi-document summarisation (MTEMDS). Such approach relies on the extraction and grouping of arbitrary length sentences from the corpus, where each group consists of similar sentences representing a topic [Moens et al., 1999, McKeown et al., 1999, Radev et al., 2000, Hardy et al., 2002, Sarkar, 2009, El-Ghannam and El-Shishtawy, 2013, Cao et al., 2017]. While a subset of these approaches relies on supervised classification methods to perform such grouping [Cao et al., 2017], other approaches rely on unsupervised clustering methods [Moens et al., 1999, McKeown et al., 1999, Radev et al., 2000, Hardy et al., 2002, Sarkar, 2009, El-Ghannam and El-Shishtawy, 2013] where sentence similarities are used as a ranking measure.

2.3.3 Tracking Topics in Dialogues

As seen in previous chapters, DSMs have enabled the discovery of topics in sentences and documents by mapping them into semantic spaces. A particularly relevant application of topic identification is in *dialogue systems* where requests made by users are carried out and answered based on the topic of the request [Jokinen et al., 2002] (Table 6.1). Such topics are used in downstream dialogue managers to updates their internal state, while also sending queries to knowledge bases to generate appropriate responses to users [Papangelis et al., 2017]. As such, *dialogue topic tracking* is an on-going area of research aiming at identifying the topic of each dialogue turn (i.e. utterance) in a dialogue session. In most setting, however, the topics are limited to a static set of alternatives (e.g. HOTEL or RESTAURANT) that have to be known a priori to improve on performance [Lane et al., 2007]. The limited topics coverage of such systems has proven to be a challenge for inexperienced users as they do not necessarily know in advance what topics the system is able to handle efficiently. Such users may attempt to formulate utterances that cannot be handled. At the same time, finer-grained natural language queries has gained a lot of interest in recent years in information retrieval [Di Buccio et al., 2014]. Therefore, there is an opportunity to improve the current state-of-the-art in dialogue topic tracking by bringing the two approaches together.

Yet, additional constraints render the identification of topics in dialogue utterances more challenging. Utterances are often short (e.g. "thank you" or "yes") and do not convey large amount of information on their own. Therefore one should account for long-term dependencies between utterances to provide useful background information of the conversation. Furthermore, because dialogue systems operates with real-time constraints, the topic inference has to be performed within a short time frame. These requirements call for sensible adaptations or consolidations of existing DSMs.

A simple way to track topics in dialogues is to use keyword-matching approaches. PyDial [Ultes et al., 2017] classifies user utterances into domains based on a predefined list of keywords. For instance, in the annotator utterance "I am looking for a hotel room where dogs are allowed", the term "hotel" is clearly specified and mapped by PyDial to the topic class "HOTEL". If PyDial is not able to identify the initial topic, it creates a meta-dialogue with the user until the initial topic has been identified or a maximum of number of trials has been reached. When a topic has been identified, it will continue to be used until a new topic is identified. Hence not every user utterance must contain relevant keywords. Although simple, PyDial does not leverage all the information contained in the current or previous user utterances and as such, may miss significant cues indicating a topic. Furthermore, PyDial does not take advantage of the distributional statistics of words in the dialogue history. Lagus and Kuusisto [2002] reformulate the dialogue topic tracking task as a statistical learning problem. They propose a clustering approach based on a SOM to produce topically ordered document maps of dialogue transcripts (see Section 2.1.3 for more detail). The approach have been shown to work reliably for long utterances, but has difficulties capturing shorter utterances. In such case, the authors acknowledge the importance of including the history prior to short utterances to improve on topic identification. More recently, Kim et al. [2016] proposed a model that takes as input an utterance and performs a classification of its topic accounting for the dialogue history up to that particular utterance. The architecture is based on a recurrent convolutional neural networks (RCNN) [Donahue et al., 2017, Karpathy and Fei-Fei, 2015, Vougiouklis et al., 2016], that is, a composition of a CNN [LeCun et al., 1999] and an LSTM [Hochreiter and Schmidhuber, 1997]. While a CNN computes a fixed-size feature vector for each utterance, the LSTM captures the dependency in time between the feature vectors. Now, such models learn to classify utterances into topics that have to be known a priori, thus failing Aim 3. We address this shortcoming in Chapter 6 where we extend the Kim et al. [2016] model to support open-domain topic tracking where identification is not restricted to a limited set of topics.

Finally, a number of other approaches to track topics in dialogues rely on topic models [Guo et al., 2018, Khatri et al., 2018]. For example, attentional deep average network (ADAN) [Guo et al., 2018] extends DAN (Section 2.1.3) in conjunction with a topic model to jointly classify the topic of each utterance, and identify the topic-specific keywords in each utterance that relate the most to its topic. For example, the utterance "I am looking for a hotel room where dogs are allowed" will be classified as "HOTEL", and the words "hotel" and "room" identified as topic-specific keywords. This approach allows a finer-grained understanding of the topics and relevant keywords in each utterance. However, ADAN does not use past utterances for more accurate predictions. To address this, contextual attentional deep average network (CADAN) [Khatri et al., 2018] augments the supervised topic models used in ADAN by incorporating features (i.e. dialog acts) that capture conversational context.

2.4 Summary

In this chapter, we have outlined some of the main works and approaches used to compute word, sentence and document representations for the purpose of topic identification. These representations have been the crucial breakthrough that led to the recent surge of machine learning research in NLP. In particular, we reviewed prediction-based models which compute embeddings based on language models in Section 2.1, and count-based models which leverage global co-occurrence statistics of words in Section 2.2. By discovering latent semantics in text, DSMs holds great potential for topic extraction.

In the later sections we have identified a number of opportunities and challenges offered by DSMs. Another promising direction is the tracking of topic in dialogues. In such a real-world application, efficiency and accuracy are important factors to ensure user engagement. However both short utterances and real-time requirements make the reliable learning of topics challenging in this setting. We have also established that it is essential for DSMs to learn topic associations to documents that align with human judgment. We identified one evaluation method in Section 2.3.1 where non-expert are asked to identify the topics that are not relevant to the documents.

In the next chapter, we evaluate scalable methods that do not require human expertise to assess the the relevance of topics to documents, when such documents have either a single or multiple topics.

Chapter 3

Evaluating Topic Classification by Aggregating Human Judgments

As per Aim 1 outlined in Chapter 1, we wish to evaluate the relevance of the topics assigned by topic models. In this chapter we partially satisfy Aim 1 by first reviewing existing literature and key algorithms relevant to combining judgments from multiple annotators when documents have a single topic, and show that these algorithms are not suited to assess topic models. In so doing, we also provide the basis for our contribution presented in Chapter 4 which address the problem of aggregating judgments for documents having multiple topics (Aim 1). More specifically, in Section 3.1, we review methods that do not account for the annotators' accuracy in the aggregation. In Section 3.2, we review methods that do take into account the annotators' accuracy by means of a single weight. In Section 3.3 we review methods which weigh the annotators using confusion matrices. Based on this analysis, we identify confusion matrix-based models, and in particular IBCC, as the most promising point of departure for our work. We give a formal description of IBCC in Section 3.4.1 and Section 3.4.2. We then present two algorithms for inferring the aggregated classification from IBCC's posterior distribution, namely, Gibbs' sampling [Stuart and Geman, 1984] in Section 3.4.3, and variational Bayesian methods in Section 3.4.4. Next, we describe the generative process when documents have multiple topics in Section 3.4.5. In Section 3.4.6, we give a step-by-step example to illustrate how IBCC fails to aggregate judgments from documents with multiple topics accurately. Finally, we evaluate IBCC empirically with synthetic data in Section 3.4.7. The chapter concludes with a summary discussing the algorithms presented in this chapter, and the areas where new work is needed to address IBCC's limitations in order to meet Aim 1 outlined in Chapter 1.

3.1 Non-weighted Aggregation Methods

In this section we review a number of non-weighted aggregation methods that do not account for the annotators' accuracy. Among these, the *product rule* and the *sum rule* are significant because they are the basis of more complex methods which will be detailed in Section 3.2. In particular, the product rule aggregates the individual annotators' judgments using a geometric average. These judgments are expressed by the annotators in terms of probabilities. The product rule can be derived from Bayes' theorem by assuming that the annotators are conditionally independent. For each document, we assume there exists a *target topic* t, which represents the correct judgment for that document that we wish to infer. Specifically, each annotator k provides a probability distribution $\mathbf{p}^{(k)} = \left\{p_1^{(k)}, \cdots, p_J^{(k)}\right\}$ where $p_j^{(k)} = p\left(t = j | p_j^{(k)}\right)$ is the probability that the target topic t has value j among J alternatives. Using Bayes' theorem, we can write the posterior probability of t given a set of judgments $\mathbf{p}_j = \left\{p_j^{(1)}, \cdots, p_j^{(K)}\right\}$ for the category j from K annotators as

$$\begin{aligned} p\left(t=j|\mathbf{p}_j\right) &= p\left(t=j\right)\frac{p\left(\mathbf{p}_j|t=j\right)}{p\left(\mathbf{p}_j\right)} \\ &= p\left(t=j\right)\prod_k \frac{p\left(p_j^{(k)}|t=j\right)}{p\left(p_j^{(k)}\right)}. \end{aligned}$$

Since $\frac{p(a|b)}{p(a)} = \frac{p(b|a)}{p(b)}$ by definition, this gives,

$$\begin{aligned} p\left(t=j \mid \mathbf{p}_j\right) &= p\left(t=j\right)\prod_k \frac{p\left(t=j|p_j^{(k)}\right)}{p\left(t=j\right)} \\ &= p\left(t=j\right)^{1-K}\prod_k p_j^{(k)}. \end{aligned} \tag{3.1}$$

The product rule can then be used to determine the most likely value $\hat{t}$ of the target topic t,

$$\hat{t} = \arg\max_j \left(p\left(t=j\right)^{1-K}\prod_k p_j^{(k)}\right).$$

where $\arg\max_x f\left(x\right)$ is the value of x for which $f\left(x\right)$ is maximal. Note that the posterior distribution of the target topic (Equation 3.1) may be multimodal, which may impose a more sophisticated decision strategy [Farrell and Mammone, 1995]. Furthermore, if an annotator assigns a probability of zero, the posterior probability will also be zero regardless of other annotators' judgments. Hence, an individual annotator has the unfortunate capability of vetoing against a topic.

In contrast, the sum rule aggregates the individual annotators' judgments using the arithmetic average

$$p\left(t=j|\mathbf{p}_j\right)=\frac{1}{K}\sum_k p_j^{(k)}, \tag{3.2}$$

from which we can then infer the target topic according to

$$\hat{t}=\arg\max_j\left(\sum_k p_j^{(k)}\right).$$

In contrast to the product rule, the sum rule is less affected by *mis-calibrated probabilities* provided by the annotators. In this context, a set of probabilities $\mathbf{p}^{(k)}$ is mis-calibrated if the outcomes predicted to occur with probability $p_j^{(k)}$ do not occur about $p_j^{(k)}$ fraction of the time [Naeini et al., 2015]. A typical example is an annotator providing extreme judgments (e.g. annotators assigning probability values close to either zero or one). Therefore the sum rule relies less on all annotators submitting trustworthy judgments. However, since the sum rule averages the individual distributions, this decreases the aggregation accuracy.

In situations where annotators are not able to provide probability distributions $\mathbf{p}^{(k)}$, the sum rule can be adapted to deal with discrete topics as input to the aggregation method. The simplest way to combine discrete topics is *plurality voting* [Shoham and Leyton-Brown, 2008] where the most frequent category among the judgments provided by each annotator is chosen. In more detail, each annotator k provides a discrete judgment $c^{(k)}\in[1,\cdots,J]$ for a document, where there are J possible categories. The aggregated judgment using plurality voting is given by

$$\hat{t}=\mathrm{argmax}_j\left(\sum_k\delta\left(j-c^{(k)}\right)\right), \tag{3.3}$$

where the *Kronecker delta* function is defined by

$$\delta(x)=\begin{cases}1 & \text{if } x=0,\\ 0 & \text{otherwise.}\end{cases} \tag{3.4}$$

In other words, this method selects the topic that receives the largest number of judgments if the solution is unique. In order to alleviate the problem of ties, we can either ensure an odd number of annotators, randomly select a topic among the most likely candidates, or hold a final voting round among the tied topics [Shoham and Leyton-Brown, 2008]. Variants of plurality voting introduce a minimum threshold h of judgments above which a candidate category is selected

$$\hat{t}=\left\{j:\sum_k\delta\left(j-c^{(k)}\right)>h\right\}, \tag{3.5}$$

where

$$h = \begin{cases} \frac{K}{2} + 1 & \text{if } K \text{ is even,} \\ \frac{K+1}{2} & \text{otherwise.} \end{cases}$$

Any threshold h such that $\frac{1}{2} < \frac{h}{K} < 1$ is called *supermajority voting*, while $\frac{h}{K} = 1$ is called *unanimity.* A popular example is *majority voting* which selects the topic which received strictly more than 50% of judgments from K annotators. It should be noted that these variants do not always produce a solution. However if it does have a solution for thresholds above 50%, it is unique and it will coincide with the plurality solution. For this reason, we will favour plurality voting over majority voting in later discussions.

The advantage of the methods described in this section is that they are relatively simple, and frequently yield useful results with a moderate amount of computation. However, they assume that each annotators is equally likely to be informative. This is addressed by opinion pools and confusion matrix-based models methods in Sections 3.2 and Section 3.3 respectively.

3.2 Weighted Aggregation Methods

Improvements to the accuracy of aggregation can be obtained by learning the annotators' reliability based on the observations of the judgments they provided in the past. This is achieved by relaxing the assumption held by non-weighted aggregation methods that annotators are equally competent. Perhaps the most common methods fall into the category known as *opinion pools.* The two main types, namely linear and logarithmic, are discussed in the remainder of this section.

3.2.1 Linear Opinion Pools

The *linear opinion pool* (LinOp) [Bacharach, 1979] (also known as *finite mixture distribution* in the statistical literature [Ravat, 2008]) is a weighted arithmetic average of the individual distributions $\mathbf{p}^{(k)}$ provided by each annotator k

$$p\left(t = j | \mathbf{p}_j\right) = \sum_k \omega^{(k)} p_j^{(k)}, \quad \forall j \in \{1, \cdots, J\} \tag{3.6}$$

where $\omega^{(k)}$ is the weight of annotator k in the aggregation. While majority voting (Equation 3.2) assumed that each annotator is equally likely to be informative of the correct topic, LinOp introduces a weight $\omega^{(k)}$ for each annotator k that represent its accuracy. For convenience, it is common to normalise the weights $\sum_k \omega^{(k)} = 1$ where $\omega^{(k)} \in [0, 1]$ such that $\sum_k \omega^{(k)} p_j^{(k)}$ is a probability distribution. A similar decision rule can be taken

$$\hat{t} = \arg\max_j \left(\sum_k \omega^{(k)} p_j^{(k)} \right).$$

When the annotators provide discrete topics, rather than probabilities, the *weighted plurality voting* is given by

$$\hat{t} = \arg\max_j \left(\sum_k \omega^{(k)} \delta \left(j - c^{(k)} \right) \right). \tag{3.7}$$

When calculating the weights in LinOp and the weighted plurality voting, two strategies are commonly sought. *Model selection* assumes that one annotator closely matches the *data generating model* (DGM), that is the (unknown) model that generated the target topics. Model selection integrates over the uncertainty as to which annotator this is. LinOp can therefore be seen as a *soft selection* method which uses weights rather than making a hard choice on a single annotator. *Bayesian model averaging* (BMA) [Leamer, 1978] and *Bayesian optimal classifiers* (BOC) are two of the main methods for soft model selection. BMA and BOC weigh each annotator by the posterior probability that it is the DGM given the documents. While BOC searches over an entire space of models with different parameters, BMA is typically applied to a sample or given set of models, often using methods such as the Gibbs' sampling algorithm to search for the correct model efficiently [Opper and Haussler, 1991]. However, if the DGM is not present in the ensemble, BMA tends to select the annotator closest to the DGM [Clarke, 2003], so is not appropriate if we wish to improve on the best annotator by combining him with others [Minka, 2000]. It is also unclear how BMA should handle judgments from annotators that are known to be unreliable. To take full advantage of an ensemble, finding the most accurate annotator is not enough. The error reduction advantages that the combination of annotators brings by directly inferring the optimal combination need also to be leveraged. Furthermore, the classifiers that match the DGM might not be present in the ensemble. *Model combination* aims at uncovering the best possible combination of annotators from the space of possible combinations. Model combination can outperform soft selection models when none of the annotators follows the DGM [Clarke, 2003]. In particular, methods such as *stacked generalisation* [Wolpert, 1992] assesses the error rate of each annotator using a technique based on cross validation. This involves testing each annotator on documents they have not previously seen, where the assessor knows the correct target topic. The annotators' judgments for any new documents can be combined by any method that takes these error rates into account, such as LinOp. Building on this idea, *Bayesian model combination* (BMC), a modified version of BMA, augments the space of potential models with combinations of annotators and achieves better empirical results than BMA (see Monteith et al. [2011] for an in-depth comparison). In principle, BMC is equivalent to using BMA on an enriched space containing all possible combinations of models. In practice, however, this space is large and requires sampling [Monteith et al., 2011].

3.2.2 Logarithmic Opinion Pool

As discussed in Section 3.1, a key drawback of the product rule (Equation 3.1) is that it does not account for mis-calibrated annotators. The *logarithmic opinion pool* (LogOp) addresses this problem by relaxing the assumption that annotators provide accurate judgments by introducing weights to calibrate the annotators' judgments, so that annotators whose responses appear over-confident or overly uncertain are corrected [Lindley et al., 1979]. The LogOp is the normalised weighted geometric mean of the individual judgments, that is

$$\hat{p}\left(t=j|\mathbf{p}_j\right)=\frac{1}{K}\prod_k\left(p_j^{(k)}\right)^{w_k}, \tag{3.8}$$

where $K=\sum_\iota\prod_k\left(p_\iota^{(k)}\right)^{w_k}$ is a normalising constant (see [Heskes, 1998] for the proof). Unlike LinOP, the weights are not constrained to be non-negative or to sum to one [Kahn, 2004]. A number of methods have been proposed to assign the weights w_k in LogOp [Genest and Zidek, 1986, Heskes, 1998, Benediktsson and Swain, 1992, Kahn, 2004]. *Discriminative* approaches, such as *logistic regression* [Kahn, 2004], estimate the weights directly from $\hat{p}\left(t=j|\mathbf{p}_j\right)$ (Equation 3.8) through optimisation. Alternatively, *generative* approaches (also referred to as *supra-Bayesian methods* [Lindley et al., 1979]) first define a joint distribution $\hat{p}\left(t=j,\mathbf{p}_j\right)$ over the target topics and the annotators' judgment, and then uses Bayes' theorem to obtain the posterior probability of the target topics given the judgments, that is,

$$\hat{p}\left(t=j|\mathbf{p}_j\right)=\frac{p\left(\mathbf{p}_j|t=j\right)}{p\left(\mathbf{p}_j\right)}p\left(t=j\right). \tag{3.9}$$

The term "generative" derives from the fact that $\hat{p}\left(t=j,\mathbf{p}_j\right)$ can also be transformed into $\hat{p}\left(\mathbf{p}_j|t=j\right)$ to generate synthetic judgments. Equation 3.9 can then be used for decision making by, for example, selecting the most probable target topic

$$\hat{t}=\arg\max_j p\left(t=j|\mathbf{p}_j\right). \tag{3.10}$$

The denominator $p\left(\mathbf{p}_j\right)$ in Equation 3.9 does not depend on t and can be ignored, giving

$$\hat{t}=\arg\max_j p\left(\mathbf{p}_j|t=j\right)p\left(t=j\right).$$

The judgments $\mathbf{p}_j$ are considered as observed data from an unknown multidimensional distribution over possible judgments and target topics. As this distribution is unknown, we may approximate it with a distribution $p\left(\mathbf{p}_j|t=j\right)$ (also called a *likelihood function*) parameterised by unknown parameters, the weights $\mathbf{w}$. This is then combined with the prior belief of the target topic $p\left(t=j\right)$. If the decision rule is to select the target topic that maximises the posterior distribution (Equation 3.10), we can ignore the constant term $p\left(\mathbf{p}_j\right)$ (also called *marginal likelihood* [Aldrich and others, 1997] or *model evidence*

[Friel and Wyse, 2012]) on the grounds that $\arg\max_x f(\alpha x) = \arg\max_x f(x)$ for any constant $\alpha \in \mathbb{R}$. Dependences between annotators can be induced by the choice as likelihood function. However, defining an accurate full joint likelihood function over the annotators' judgment is challenging as it would either require a vast amount of data or much guesswork. Furthermore, evaluating it would be computationally expensive. For these reasons, approximations are usually preferred, leading to the well known naïve Bayes classifier. This approach assumes that the annotators' judgments are independent conditioned on the target topic, therefore ignoring any correlations between annotators. The resulting likelihood function is given by

$$p\left(\mathbf{p}_j | t = j\right) = \prod_k p\left(p_j^{(k)} | t = j\right). \tag{3.11}$$

The choice of the likelihood is often arbitrary. For example, the *Gaussian LogOp* [Kahn, 2004] assumes that the log-odds of the likelihood of the annotators' judgments is distributed according to a Normal distribution, that is, $p\left(\mathbf{p}_j | t = j\right) = \prod_k N\left(\frac{p_j^{(k)}}{1-p_j^{(k)}} \middle| t = j\right)$. Despite this strong assumption, naïve Bayes approaches produce reliable decisions when annotators are not biased, that is, when annotators do not consistently and predictably favour one judgment over others. Bias would reduce the contribution of an honest but biased annotator in the aggregation.

3.3 Confusion Matrix-based Generative Models

Although opinion pools capture the accuracy of the individual annotators by way of a single weight per annotator, the weights alone are not sufficient to measure their quality. Specifically, an annotator may be careful but biased, giving consistently and predictably incorrect answers that can be recovered [Ipeirotis et al., 2010, Welinder et al., 2010].

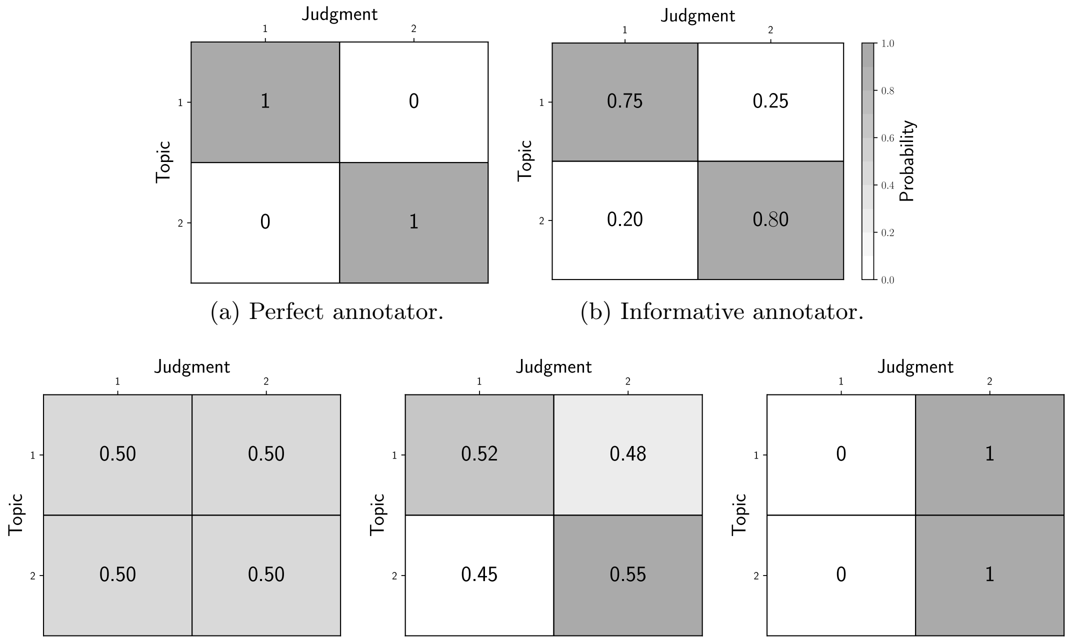
Example 3.1. *Consider two annotators providing judgments regarding the topic of a document with two alternatives: fiction and non-fiction. The first annotator is always incorrect: labels fiction documents as non-fiction, and non-fiction as fiction. The second annotator classifies all documents as fiction irrespectively of their topic. A simple accuracy analysis indicates that the error rate of the first annotator is 100%, while the error rate of second annotator is only 50%. However, the errors of the first annotator are easily reversible, while the errors of the second annotator are not. Thus, the first annotator is informative, while the second annotator is not.*

To capture such characteristics, *confusion matrices* provide a more detailed assessment than the accuracy alone (i.e. the proportion of correct classification). In more detail, a confusion matrix $\mathbf{\Pi}^{(k)}$ for annotator k is a square stochastic matrix of dimension $(J \times J)$ capturing the probabilistic dependency between the annotator's responses and the target topic among J alternatives

$$\mathbf{\Pi}^{(k)} = \begin{pmatrix} \pi_{1,1}^{(k)} & \cdots & \pi_{1,J}^{(k)} \\ \vdots & \ddots & \vdots \\ \pi_{J,1}^{(k)} & \cdots & \pi_{J,J}^{(k)} \end{pmatrix} = \begin{pmatrix} \vdots \\ \boldsymbol{\pi}_j^{(k)} \\ \vdots \end{pmatrix}. \tag{3.12}$$

Each row $\boldsymbol{\pi}_j^{(k)}$ represents a possible topic $j \in \{1, \cdots, J\}$, while each column represents the annotator's possible judgment $l \in \{1, \cdots, J\}$. The probabilities $\pi_{j,l}^{(k)} = p\left(c_i^{(k)} = l | t_i = j, \boldsymbol{\pi}_j^{(k)}\right)$ for $j \in \{1, \cdots, J\}$ and $l \in \{1, \cdots, J\}$ are called individual *error-rates* for the k-th annotator (although this set contains $\pi_{j,j}^{(k)}$ for $j \in \{1, \cdots, J\}$, which are the probabilities that the annotator provides the true topic). Note that the error-rates are conditional probabilities where $\sum_{l=1}^{L} \pi_{j,l}^{(k)} = 1$. This representation makes it easy to see the annotator's misjudgment of a given topic.

We can separate annotators into five categories depending on the profile of their confusion matrix: perfect, informative, random, semi-random, and uniform [Vuurens et al., 2011]. While perfect annotators have the identity matrix as confusion matrix (i.e. $\mathbf{\Pi} = \mathbf{I}$) (Figure 3.1a), informative annotators try to the best of their abilities to complete the tasks. As such, informative annotators are typically bias toward the consensus topic but may be imprecise in their judgments. On the other hand, random, semi-random and uniform annotators intentionally undermine the quality of the aggregation by providing erroneous judgments. They are collectively referred to as *spammers*. Uniform spammers (Figure 3.1e) use a fixed uniform judgment pattern across all documents (i.e. $\boldsymbol{\pi}_j^{(k)} = \delta\left(l^*\right)$ for some fixed $l^* \in \{1, ..., J\}$). Random spammers (Figure 3.1c) provide unique meaningless answers for each document (i.e. $\boldsymbol{\pi}_j^{(k)} = \text{Uniform}\left(c_i^{(k)} = l\right)$) and semi-random spammers (Figure 3.1d) also answer a few questions properly.



(a) Perfect annotator.

(b) Informative annotator.

(c) Random spammer.

(d) Semi-random spammer.

(e) Uniform spammer.

Figure 3.1: Illustrative examples of confusion matrices $\mathbf{\Pi}^{(k)}$ for different type of annotators in a binary classification scenario. The values $\pi_{j,l}^{(k)}$ for informative annotators and semi-random spammers were chosen arbitrarily for illustrative purposes.

A number of models use confusion matrices to aggregate judgments. In particular, the Dawid and Skene model (DS) [Dawid and Skene, 1979] is a parametric generative aggregation model which builds on the naïve Bayes assumption (Section 3.2.2) to infer the confusion matrix of each annotator and the target topics simultaneously using the expectation maximisation (EM) algorithm. This is achieved by: (i) estimating the target topic by weighing the judgments of each annotator according to the current estimates of their competence (as given by the confusion matrix), and (ii) re-estimating the confusion matrices based on the current beliefs about the target topic of each document. Since its introduction, DS has been extended in a number of ways. In particular, IBCC extends DS by introducing Dirichlet prior distributions over the topic proportions and the rows of the confusion matrices. This enables more flexibility in incorporating prior background information about the annotators. Going a step further, community-based Bayesian classifier combination (CBCC) [Venanzi et al., 2014] relaxes the assumption of independency between annotators in IBCC by introducing the concept of latent annotator communities. Each community is associated with a confusion matrix, which represents the average confusion matrix of its members. In contrast to the standard IBCC where the annotators' confusion matrix are independent, CBCC models the correlation between the confusion matrices. CBCC has proven to be useful in situations where each annotator does not provide enough judgments to reliably estimate their quality. Furthermore, each community can be assigned a subset of documents to judge depending on the suitability of their annotators. The number of communities is a parameter of the model and needs to be specified prior to the inference. However the number of annotators required by CBCC to accurately infer the communities is larger compared to IBCC due to the additional parameters required to infer such communities. As IBCC is the basis of our contribution in Chapter 4, we give a detailed discussion of this model in the next section.

3.4 Independent Bayesian Classifier Combination

In this section, we first give a formal description of IBCC and specify its relationship with the earlier DS model in Section 3.4.1. We then present the Gibbs' sampling algorithm [Stuart and Geman, 1984] in Section 3.4.3 and variational Bayesian methods in Section 3.4.4 which enable IBCC to perform inference and deduce the aggregated classification for documents with single topic from IBCC's posterior distribution. Since one of the aims of our work is to evaluate the topic assignment provided by topic models (Aim 1), we derive a generative process when documents have multiple topics in Section 3.4.5. In Section 3.4.6, we illustrate how IBCC handles such a scenario with a step-by-step example, and give a more thorough empirical evaluation in Section 3.4.7.

3.4.1 Model Description

Assume there is a set of I documents for which we want to infer a set of latent topics $\mathbf{t} = \{t_1, \cdots, t_I\}$, where there is a single topic per document (i.e. the target topic). The target topic t_i for document i takes a value in $j \in \{1, \cdots, J\}$, where J is the number of alternatives. Topics are assumed to be drawn from a categorical distribution with probability

$$t_i|\boldsymbol{\kappa} \sim \mathrm{Cat}\left(\boldsymbol{\kappa}\right). \tag{3.13}$$

Given a set of K annotators, each annotator $k \in \{1, \cdots, K\}$ submits a judgment $c_i^{(k)} = l$ of the target topic $t_i = j$ for document i, where $l \in \{1, \cdots, J\}$ is the set of discrete judgments that the annotator can make. A judgment $c_i^{(k)}$ from annotator k is assumed to have been drawn from a categorical distribution, taking value l with probability

$$c_i^{(k)} = l|\boldsymbol{\pi}_j^{(k)} \sim \mathrm{Cat}\left(c_i^{(k)} = l|\boldsymbol{\pi}_j^{(k)}\right) \tag{3.14}$$

Furthermore, the annotators' judgments are assumed to be conditionally independent given the target topic t_i

$$c_i^{(k)} \perp c_i^{(\{1,\cdots,K\}\backslash k)}|t_i, \qquad \forall k \in \{1, \cdots, K\}.$$

This is the assumption commonly used in *naïve Bayes classifiers* [Frank and Bouckaert, 2006] which ignores any correlations between annotators. It is a reasonably good assumption on crowdsourcing platforms since annotators do not typically interact with each other. The confusion matrix $\boldsymbol{\Pi}^{(k)} = \left\{ \boldsymbol{\pi}_j^{(k)} : j = 1, \cdots, J \right\}$ for annotator k has dimension $(J \times J)$. All rows of the confusion matrix are assumed independent within and across annotators

$$\boldsymbol{\pi}_i^{(k)} \perp \boldsymbol{\pi}_j^{(\{1,\cdots,K\}\setminus k)}, \qquad \forall k \in \{1, \cdots, K\} \text{ and } \forall i \neq j.$$

This means that an annotator's ability to identify a given topic is not dependent on his ability to identify the other alternatives. The generative process we have described so far, that is, the random process by which IBCC assumes the annotators' judgment $\mathbf{c}$ arose, is summarised in Algorithm 1.

Algorithm 1 Generative process expressing hypotheses about the way in which the annotators' judgment for each documents have been generated.

```
Input: model's parameters (i.e. topic proportion $\boldsymbol{\kappa}$ and confusion matrices $\boldsymbol{\Pi}^{(k)}$)

for each document $i \in \{1, \cdots, I\}$ do                          ▷ for each document $i$
    Sample $t_i \sim \text{Categorical}(\boldsymbol{\kappa})$                          ▷ sample the target topic
    for each annotator $k \in \{1, \cdots, K\}$ do                     ▷ for each annotator $k$
        Sample $c_i^{(k)} \sim \text{Categorical}\left(\boldsymbol{\pi}_{t_i}^{(k)}\right)$    ▷ sample the annotator's judgment of $t_i$
    end for
end for

return c
```

When the parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\Pi}$ are fixed and unknown, the model described so far correspond to the DS model from which IBCC is a Bayesian extension (see next section). The associated factor graph of the Dawid and Skene model is shown on Figure 3.2.

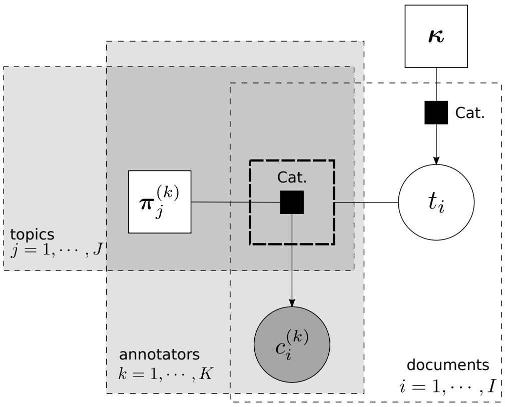


Figure 3.2: Factor graph of the DS model. The parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\Pi}$ are point values conforming to non-Bayesian statistics.

3.4.2 Bayesian Inference

IBCC introduces the Bayesian formalism to DS and considers the parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\Pi}$ as random variables with given prior distributions and hyperparameters $\boldsymbol{\nu}$ and $\mathbf{A}$ (i.e. the parameters of the prior distributions). Therefore, in line with Bayesian modelling, we assign a conjugate Dirichlet prior distribution to the parameter vector $\boldsymbol{\kappa} = [\kappa_1, \cdots, \kappa_j]$, such that

$$\boldsymbol{\kappa}|\boldsymbol{\nu} \sim \mathrm{Dir}\left(\boldsymbol{\nu}\right). \tag{3.15}$$

The prior in Equation 3.15 is parameterised by a vector $\boldsymbol{\nu} = [\nu_1, \cdots, \nu_j]$ of dimensions J – one for each of the possible topic. Intuitively, we can view the hyperparameter $\boldsymbol{\nu}$ as *pseudo-counts* of prior observations, that is, the number of documents in each topic that the annotators have already judged (see Appendix A for a discussion on Dirichlet distributions). A conjugate Dirichlet prior distribution is similarly introduced over the parameter $\boldsymbol{\pi}_j^{(k)}$ with hyperparameter $\boldsymbol{\alpha}_j^{(k)}$ such that

$$\boldsymbol{\pi}_j^{(k)}|\boldsymbol{\alpha}_j^{(k)} \sim \mathrm{Dir}\left(\boldsymbol{\pi}_j^{(k)}|\boldsymbol{\alpha}_j^{(k)}\right). \tag{3.16}$$

The hyperparameters $\boldsymbol{\alpha}_j^{(k)}$ form a matrix $\mathbf{A}^{(k)} = \left\{\boldsymbol{\alpha}_1^{(k)}, \cdots, \boldsymbol{\alpha}_J^{(k)}\right\}$, where each row j is a point value vector $\boldsymbol{\alpha}_j^{(k)}$. The hyperparameter $\mathbf{A}$ can be chosen to represent any prior level of uncertainty in the values of the annotators' confusion matrix, and can be regarded as pseudo-counts of prior observations. In this chapter we will not attempt to optimise the value of the hyperparameters $\boldsymbol{\nu}$ and $\mathbf{A}$, often referred as *hyperparameter optimisation* or *model selection* [Rasmussen and Williams, 2006]. This can be deduced from a training set or from expert judgment. Additionally, as the size of the observations becomes large, any reasonable choices of prior distributions will only have a minor effect on posterior inferences ["Bernstein-von Mises theorem", Vaart, 2000]. The resulting factor graph summarising IBCC's set of assumptions is depicted in Figure 3.3. The joint distribution of IBCC is defined over all the random variables in the model (i.e. $\mathbf{t}$, $\mathbf{c}$, $\boldsymbol{\kappa}$ and $\boldsymbol{\Pi}$).

Proposition 3.1. *The joint distribution of IBCC is given by*

$$\begin{aligned} p\left(\mathbf{t}, \mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}; \boldsymbol{\nu}, \mathbf{A}\right) &= p\left(\mathbf{t}, \mathbf{c}|\boldsymbol{\kappa}, \boldsymbol{\Pi}\right) p\left(\boldsymbol{\kappa}; \boldsymbol{\nu}\right) p\left(\boldsymbol{\Pi}; \mathbf{A}\right) \\ &= \prod_{i=1}^{I}\left\{\kappa_{t_i}\prod_{k=1}^{K}\pi^{(k)}_{t_i, c_i^{(k)}}\right\} \times \\ &\quad \frac{1}{\mathrm{B}\left(\boldsymbol{\nu}\right)}\prod_{j=1}^{J}\kappa_j^{\nu_j - 1} \times \\ &\quad \prod_{k=1}^{K}\prod_{j=1}^{J}\left\{\frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}_j^{(k)}\right)}\prod_{l=1}^{L}\left(\pi_{j,l}^{(k)}\right)^{\alpha_{j,l}^{(k)} - 1}\right\}. \end{aligned} \tag{3.17}$$

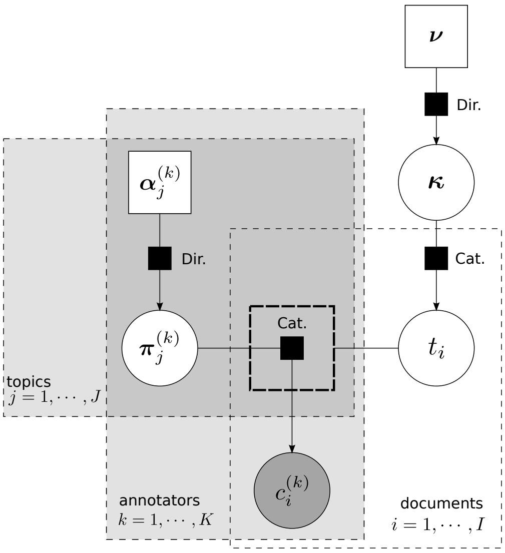


Figure 3.3: Factor graph of IBCC.

Proof. By repeated application of the Bayes theorem, the joint distribution can be written as a product of marginal and conditional distributions

$$\begin{aligned} p\left(\mathbf{t}, \mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}; \boldsymbol{\nu}, \mathbf{A}\right) &= p\left(\mathbf{t}, \mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}; \cancel{\boldsymbol{\nu}}, \cancel{\mathbf{A}}\right) p\left(\boldsymbol{\kappa} | \cancel{\boldsymbol{\Pi}}; \boldsymbol{\nu}, \cancel{\mathbf{A}}\right) p\left(\boldsymbol{\Pi}; \cancel{\boldsymbol{\nu}}, \mathbf{A}\right) \\ &= p\left(\mathbf{t}, \mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}\right) p\left(\boldsymbol{\kappa}; \boldsymbol{\nu}\right) p\left(\boldsymbol{\Pi}; \mathbf{A}\right). \end{aligned} \tag{3.18}$$

It should be noted that by making this *causal* decomposition (that is, the decomposition that replicates the data generation process), we have implicitly chosen a particular ordering, and had we chosen a different ordering we would have obtained a different decomposition and hence a different representation [Koller and Friedman, 2009]. With this in mind, the conditional probability of $\mathbf{t}$ and $\mathbf{c}$ given $\boldsymbol{\kappa}$ and $\boldsymbol{\Pi}$ can be written

$$\begin{aligned} p\left(\mathbf{t}, \mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}\right) &= p\left(\mathbf{t} | \boldsymbol{\kappa}, \cancel{\boldsymbol{\Pi}}\right) p\left(\mathbf{c} | \mathbf{t}, \cancel{\boldsymbol{\kappa}}, \boldsymbol{\Pi}\right) \\ &= p\left(t_1, \cdots, t_I | \boldsymbol{\kappa}\right) p\left(\mathbf{c}_1, \cdots, \mathbf{c}_I | t_1, \cdots, t_I, \boldsymbol{\Pi}\right) \\ &= \prod_{i=1}^{I} \left\{ p\left(t_i | \boldsymbol{\kappa}\right) p\left(\mathbf{c}_i | t_i, \boldsymbol{\Pi}\right) \right\}. \end{aligned}$$

Since $c_i^{(k)} \perp c_j^{(k)} | t_i$ for $i \neq j$, we get

$$\begin{aligned} & \prod_{i=1}^{I} \left\{ p\left(t_i | \boldsymbol{\kappa}\right) \prod_{k=1}^{K} p\left(c_i^{(k)} | t_i, \boldsymbol{\pi}_j^{(k)}\right) \right\} \\ = & \prod_{i=1}^{I} \left\{ \kappa_{t_i} \prod_{k=1}^{K} \pi^{(k)}_{t_i, c_i^{(k)}} \right\}. \end{aligned} \tag{3.19}$$

Similarly, the marginal probability of $\mathbf{\Pi}$ can be written

$$p\left(\mathbf{\Pi};\mathbf{A}\right) = \prod_{k=1}^{K} p\left(\mathbf{\Pi}^{(k)};\mathbf{A}^{(k)}\right).$$

Since $\boldsymbol{\pi}_i^{(k)} \perp \boldsymbol{\pi}_j^{(k)}$ for $i \neq j$, we get

$$\begin{aligned} & \prod_{j=1}^{J}\prod_{k=1}^{K} p\left(\boldsymbol{\pi}_j^{(k)};\boldsymbol{\alpha}_j^{(k)}\right) \\ = & \prod_{k=1}^{K}\prod_{j=1}^{J}\frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}_j^{(k)}\right)}\prod_{l=1}^{J}\left(\pi_{j,l}^{(k)}\right)^{\alpha_{j,l}^{(k)}-1}, \end{aligned} \tag{3.20}$$

where $\mathrm{B}\left(\boldsymbol{\alpha}_j^{(k)}\right) = \frac{\prod_{l=1}^{J}\Gamma\left(\alpha_{j,l}^{(k)}\right)}{\Gamma\left(\sum_{l=1}^{J}\alpha_{j,l}^{(k)}\right)}$ the multinomial multivariate Beta function and $\frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}_j^{(k)}\right)}$ is a normalising constant. Therefore, combining Equation 3.19, Equation 3.15, and Equation 3.20 into Equation 3.18 completes the proof. Alternatively, the joint distribution in Equation 3.17 can simply be deduced from the factor graph in Figure 3.3 by multiplying the local conditional distributions found in Equations 3.13, 3.14, 3.15 and 3.16. □

Since in the Bayesian setting no distinction is made between latent variables and model parameters, we can treat them on equal footing. Therefore to simplify the presentation, we denote by $\mathbf{H} = \{\mathbf{t}, \boldsymbol{\kappa}, \mathbf{\Pi}\}$ the set of hidden variables including the model's parameters, and $\mathbf{V} = \{\mathbf{c}\}$ the set of visible (i.e. observed) variables. It follows that $\mathbf{X} = \{\mathbf{H}, \mathbf{V}\}$ forms the set of all random variables in the model. The objective of Bayesian inference is to jointly infer the posterior distribution of the hidden variables $\mathbf{H}$ conditioned on the visible variables $\mathbf{V}$. To do so, we marginalise the observed jugments $\mathbf{c}$ from the joint distribution in Equation 3.17, giving the *posterior distribution*

$$p\left(\mathbf{t}, \boldsymbol{\kappa}, \mathbf{\Pi}|\mathbf{c}; \boldsymbol{\nu}, \mathbf{A}\right) = \frac{p\left(\mathbf{t}, \mathbf{c}, \boldsymbol{\kappa}, \mathbf{\Pi}; \boldsymbol{\nu}, \mathbf{A}\right)}{p\left(\mathbf{c}; \boldsymbol{\nu}, \mathbf{A}\right)}, \tag{3.21}$$

or equivalently

$$p\left(\mathbf{H}|\mathbf{V}\right) = \frac{p\left(\mathbf{X}\right)}{p\left(\mathbf{V}\right)}. \tag{3.22}$$

The denominator (i.e. the marginal likelihood, or model evidence) is a constant given by

$$p\left(\mathbf{c}; \boldsymbol{\nu}, \mathbf{A}\right) = \int_{\mathbf{t}}\int_{\mathbf{\Pi}}\int_{\boldsymbol{\kappa}} p\left(\mathbf{t}, \mathbf{c}, \boldsymbol{\kappa}, \mathbf{\Pi}; \boldsymbol{\nu}, \mathbf{A}\right) d\mathbf{t} d\boldsymbol{\kappa} d\mathbf{\Pi}, \tag{3.23}$$

or equivalently

$$p\left(\mathbf{V}\right) = \int_{\mathbf{H}} p\left(\mathbf{X}\right) d\mathbf{H}. \tag{3.24}$$

The maximum a posteriori estimator (MAP) aims at finding the value of $\mathbf{t}$, $\mathbf{\Pi}$, and $\boldsymbol{\kappa}$ (or equivalently $\mathbf{H}$) that maximises the posterior distribution given in Equation 3.21.

To do this, exact inference algorithms first compute the closed-form expression of the posterior distribution by deriving the analytical form of Equation 3.21. However, since the integration in the marginal likelihood in Equation 3.23 has no closed form expression for IBCC due to the integrations in the model evidence (Equation 3.23) [Simpson, 2014], the posterior distribution must be estimated using approximation methods. A number of approximation methods have been proposed including statistical sampling (such as *Markov chain Monte Carlo* (MCMC) [Geman and Geman, 1984]) or density approximation (such as variational approximation [Jordan et al., 1999, Attias, 2000] or Laplace approximation [Laplace, 1986]). In particular, *Gibbs' sampling* and *variational Bayesian methods* have been used successfully with IBCC [Kim and Ghahramani, 2012, Simpson et al., 2013]; but other approaches could be used if appropriate.

We present how inference is performed using Gibbs' sampling algorithm [Stuart and Geman, 1984] in the next section, and using variational Bayesian inference in Section 3.4.4.

3.4.3 Inference using Gibbs' Sampling

Gibbs' sampling is an approximate random sampling inference algorithms belonging to the MCMC family. It is applicable when the posterior distribution is not known explicitly (as in IBCC) or is difficult to sample from directly, but the conditional distribution of each variable given all the over variables is known and is easy to sample from.

We first describe the principle behind Gibbs' sampling in Section 3.4.3.1. We then derive the specific Gibbs equations for IBCC in Section 3.4.3.2.

3.4.3.1 Gibbs' Sampling

Given a posterior distribution of the form $p\left(\mathbf{H}|\mathbf{V}\right)$, Gibbs' sampling is an algorithm that involves repeated sampling of each hidden variable $\mathbf{H}_i \in \mathbf{H}$ in turn using their most recent value at each iteration until convergence is observed, or a maximum number of iterations M is reached (Algorithm 2). Gibbs' sampling generates a sequence of samples $\mathbf{H}^{[0]}, \cdots, \mathbf{H}^{[m]}$ from a Markov chain over all possible states. The stationary distribution of the Markov chain is the joint distribution in Equation 3.22.

Algorithm 2 Gibbs' sampling algorithm.

```
Initialise the chain $\mathbf{H}^{[0]} = \left\{ H_1^{[0]}, \cdots, H_n^{[0]} \right\}$
for each sample $m \in \{1, \cdots, M\}$ do
    for each $i \in \{1, \cdots, n\}$ do
        Sample $H_1^{[m]} \sim p\left(H_1 | H_2^{[m-1]}, H_3^{[m-1]}, \cdots, H_n^{[m-1]}, \mathbf{V}\right)$
        Sample $H_2^{[m]} \sim p\left(H_2 | H_1^{[m]}, H_3^{[m-1]}, \cdots, H_n^{[m-1]}, \mathbf{V}\right)$
        $\vdots$
        Sample $H_n^{[m]} \sim p\left(H_n | H_1^{[m]}, H_2^{[m]}, \cdots, H_{n-1}^{[m]}, \mathbf{V}\right)$
    end for
    Let $\mathbf{H}^{[m]} = \left\{ H_1^{[m]}, \cdots, H_n^{[m]} \right\}$          ▷ sample from the posterior distribution
end for
return $\{\hat{p}\left(H_i | \mathrm{MB}\left(H_i\right)\right)$ for $1 \leq i \leq n\}$ and $\hat{p}\left(\mathbf{H} | \mathbf{V}\right)$          ▷ empirical posterior distributions
```

As it is typically the case in MCMC algorithms, each sample in the Gibbs' sampling procedure is correlated with its predecessor. To obtain independent samples, we can systematically use every m-th sample and discard the others. This is referred as *thinning*. Additionally, if we cannot specify good initial values for the chain, we need to define a *burn-in* period where early samples are discarded. This is desirable when the rate of convergence is large with respect to the total number of samples, as it avoids bias in the inferred values. The length of the burn-in period is different for each chain. In principle, if we knew the rate of convergence of the Markov chain to its stationary distribution, we could use this information to decide on the burn-in period for the output. Unfortunately, in practice we don't have a generic way of assessing this rate of convergence as it depends on the target distribution [Li, 2005]. Practically, we can only examine visually the trace plots of the samples' value versus iterations to look for evidence of when the simulation appears to have stabilised (Figure 3.5c).

3.4.3.2 Gibbs' Sampling for IBCC

To perform inference using Gibbs' sampling on IBCC, we first need to derive the Gibbs' sampling equations which are the posterior distributions of each hidden variable, that is, $\boldsymbol{\kappa}$, $\mathbf{\Pi}$ and $\mathbf{t}$. Before doing so however, it should be noted that when Gibbs' sampling is used on graphical models such as IBCC, we condition on the *Markov blanket* of each random variables rather than the entirety of the remaining variables. Because of the dependency structure of graphical models, the Markov blanket of a node is the only knowledge needed to predict the behaviour of that node. The Markov blanket $\mathrm{MB}\left(X_i\right)$ of a random variable X_i is defined as the set of random variables consisting of its parent node, its children and the parent of its children (Figure 3.4).

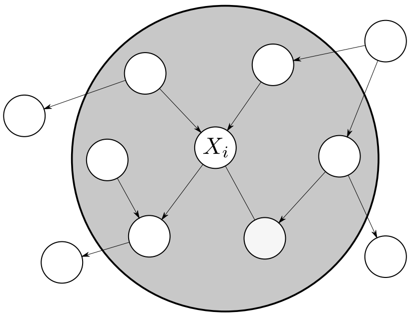


Figure 3.4: The Markov blanket of the variable X_i (shaded area) depends only on the set of its parents, children and parents of the children of that node.

Proposition 3.2. *The conditional distribution of the class proportions* $\boldsymbol{\kappa}$ *given its Markov blanket has a Dirichlet distribution given by*

$$\boldsymbol{\kappa}|MB\left(\boldsymbol{\kappa}\right) \sim Dir\left(\boldsymbol{\nu} + \boldsymbol{N}\right), \tag{3.25}$$

where $MB\left(.\right) \subset \{\mathbf{X} \cup \{\boldsymbol{\nu}, \mathbf{A}\}\}$ *is the* Markov blanket, *and the count of topic* $t_i = j$ *in* $\mathbf{t}$ *given by*

$$N_j = \left|\{i : t_i = j \text{ } for \text{ } i = 1, \cdots, I\}\right|, \tag{3.26}$$

or equivalently,

$$N_j = \sum_{i=1}^{I} \delta\left(t_i - j\right).$$

Proof. From the conditional independencies in IBCC's factor graph (Figure 3.3), the Markov blanket MB $(\boldsymbol{\kappa})$ is $\{\mathbf{t}, \boldsymbol{\nu}\}$. Noting that $\boldsymbol{\kappa}$ (Equation 3.15) is the prior of $\mathbf{t}$ (Equation 3.13), and noting that

$$\prod_{i=1}^{I} \kappa_{t_i} = \prod_{j=1}^{J} \kappa_j^{N_j}, \tag{3.27}$$

we can apply Proposition A.1 (Appendix A) to $p\left(\boldsymbol{\kappa}|\mathbf{t};\boldsymbol{\nu}\right)$ which completes the proof.

□

Proposition 3.3. *The conditional distribution of the set of error rates* $\boldsymbol{\pi}_j^{(k)}$ *for a target topic* j *and annotator* k *given its Markov blanket has a Dirichlet distribution given by*

$$\boldsymbol{\pi}_j^{(k)}|MB\left(\boldsymbol{\pi}_j^{(k)}\right) \sim Dir\left(\boldsymbol{\alpha}_j^{(k)} + \mathbf{M}_j^{(k)}\right), \tag{3.28}$$

where

$$M_{j,l}^{(k)} = \left|\left\{i : t_i = j \wedge c_i^{(k)} = l \text{ } for \text{ } i = 1, \cdots, I\right\}\right|, \tag{3.29}$$

or equivalently,

$$M_{j,l}^{(k)} = \sum_{i=1}^{I} \delta\left(t_i - j\right) \delta\left(c_i^{(k)} - l\right),$$

the number of judgments $c_i^{(k)}$ of topic l produced by annotator k for documents i with topic j.

Proof. From the conditional independencies in IBCC's factor graph (Figure 3.3), the Markov blanket MB $\left(\boldsymbol{\pi}_j^{(k)}\right)$ is $\left\{\mathbf{t}, \mathbf{c}, \boldsymbol{\alpha}_j^{(k)}\right\}$. Since the documents are independent, we get

$$p\left(\boldsymbol{\pi}_j^{(k)} | \mathbf{t}, \mathbf{c}; \boldsymbol{\alpha}_j^{(k)}\right) = \prod_{i=1}^{I} p\left(\boldsymbol{\pi}_j^{(k)} | t_i = j, c_i^{(k)}; \boldsymbol{\alpha}_j^{(k)}\right).$$

Noting that

$$\prod_{i=1}^{I} \pi^{(k)}_{t_i, c_i^{(k)}} = \prod_{l=1}^{L} \left(\pi^{(k)}_{j,l}\right)^{M^{(k)}_{j,l}} \tag{3.30}$$

and using Proposition A.1 (Appendix A) completes the proof.

□

Proposition 3.4. *The conditional distribution of the target topic t_i given its Markov blanket has a categorical distribution*

$$p\left(t_i = j | MB\left(t_i\right)\right) = \frac{\kappa_j \prod_{k=1}^{K} \pi^{(k)}_{j, c_i^{(k)}}}{\sum_{\iota=1}^{J} \kappa_\iota \prod_{k=1}^{K} \pi^{(k)}_{\iota, c_i^{(k)}}}. \tag{3.31}$$

Proof. From the conditional independencies in IBCC's factor graph (Figure 3.3), the Markov blanket MB (t_i) is $\{\mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}\}$. By definition of a marginal distribution, we get

$$\begin{aligned} p\left(t_i = j | \mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}\right) &= \frac{p\left(t_i = j, \mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}\right)}{p\left(\mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}\right)} \\ &= \frac{p\left(t_i = j, \mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}\right) \cancel{p\left(\boldsymbol{\kappa}, \boldsymbol{\Pi}\right)}}{p\left(\mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}\right) \cancel{p\left(\boldsymbol{\kappa}, \boldsymbol{\Pi}\right)}}. \end{aligned}$$

Furthermore, by introducing and marginalising t_i in the denominator we get

$$\begin{aligned} & \frac{p\left(t_i = j, \mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}\right)}{\sum_{\iota=1}^{J} p\left(t_i = \iota, \mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}\right)} \\ = & \frac{\kappa_j \prod_{k=1}^{K} \pi^{(k)}_{j, c_i^{(k)}}}{\sum_{\iota=1}^{J} \kappa_\iota \prod_{k=1}^{K} \pi^{(k)}_{\iota, c_i^{(k)}}} \\ \propto & \kappa_j \prod_{k=1}^{K} \pi^{(k)}_{j, c_i^{(k)}} \end{aligned}$$

which completes the proof.

□

Equations 3.25, 3.28 and 3.31 are known as *full conditional distributions*. Therefore, even if we do not have a closed form for the posterior distribution in Equation 3.21, we can easily sample from the Gibbs' conditional Equations 3.25, 3.28 and 3.31. Algorithm 3 describes the step-by-step Gibbs's sampling algorithm as it is applied to IBCC.

Algorithm 3 Inference using Gibbs' sampling for IBCC. We denote $\left\{\boldsymbol{\kappa}^{[m]}, \boldsymbol{\Pi}^{[m]}, \mathbf{t}^{[m]}\right\}$ the set of hidden variables at step m.

```
 1: Input: $M \gg 0$, $\mathbf{A}$, $\boldsymbol{\nu}$
 2: Initialise $\boldsymbol{\kappa}^{[0]}$, $\boldsymbol{\Pi}^{[0]}$ and $\mathbf{t}^{[0]}$          ▷ initial state of the Markov chain
 3:
 4: for each $m$ in $\{1, \cdots, M\}$ do          ▷ for each Gibbs' sample
 5:     Sample $\boldsymbol{\kappa}^{[m]} \sim p\left(\boldsymbol{\kappa} | \mathbf{t}^{[m-1]}, \boldsymbol{\nu}\right)$          ▷ sample the topic proportion
 6:
 7:     for each $k \in \{0, \cdots, K\}$ do
 8:         for each $j \in \{0, \cdots, J\}$ do
 9:             Sample $\boldsymbol{\pi}_j^{(k)[m]} \sim p\left(\boldsymbol{\pi}_j^{(k)} | \mathbf{t}^{[m-1]}, \mathbf{c}, \boldsymbol{\alpha}_j^{(k)}\right)$          ▷ sample a row of conf. matrix
10:         end for
11:     end for
12:
13:     for each $i \in \{0, \cdots, I\}$ do
14:         for each $j \in \{0, \cdots, J\}$ do
15:             $p_{i,j}^{[m]} = p\left(t_i = j | \boldsymbol{\kappa}^{[m]}, \boldsymbol{\pi}_j^{(k)[m]}, \mathbf{c}\right)$
16:         end for
17:         Sample $t_i^{[m]} \sim \text{Categorical}\left(\mathbf{p}_i^{[m]}\right)$          ▷ sample a topic
18:     end for
19:
20:     Record the samples $\left\{\boldsymbol{\kappa}^{[m]}, \boldsymbol{\Pi}^{[m]}, \mathbf{t}^{[m]}\right\}$
21: end for
22:
23: return $\hat{p}(\boldsymbol{\kappa} | \text{MB}(\boldsymbol{\kappa}))$, $\hat{p}(\boldsymbol{\Pi} | \text{MB}(\boldsymbol{\Pi}))$, $\hat{p}(\mathbf{t} | \text{MB}(\mathbf{t}))$          ▷ empirical posterior
    distributions
```

In particular, the algorithm proceeds as follows. We start by initialising the values of $\mathbf{t}$, $\boldsymbol{\Pi}$ and $\boldsymbol{\kappa}$ of the Markov chain (line 2). We then sample a value for $\boldsymbol{\kappa}$ from the distribution given in Equation 3.25, using the current value of $\mathbf{t}$ (line 5). We replace the current value of $\boldsymbol{\kappa}$ with the new sample. We proceed similarly with each row $\boldsymbol{\pi}_j^{(k)}$ of the annotator's confusion matrices from Equation 3.28 conditioned on the current value of $\mathbf{t}$ (line 9). We then replace the current value of $\boldsymbol{\pi}_j^{(k)}$ with the new samples. A sample of t_i for each document i is drawn from Equation 3.31 using current values of $\boldsymbol{\Pi}$ and $\boldsymbol{\kappa}$ (line 14). The observed values of $\mathbf{c}$ are fixed and are not sampled. We store the current sample of the posterior distribution in a list of samples (line 17) to build the empirical posterior distributions (line 20). Finally, the mean or mode of each variable (i.e. $\overline{\boldsymbol{\kappa}}$, $\overline{\boldsymbol{\Pi}}$ and mode($\mathbf{t}$)) is taken as estimators for each variable, such that,

$$\overline{\boldsymbol{\kappa}} = \frac{1}{M}\sum_{m}\boldsymbol{\kappa}^{[m]}, \tag{3.32}$$

and

$$\overline{\boldsymbol{\pi}_j^{(k)}} = \frac{1}{M}\sum_{m}\boldsymbol{\pi}_j^{(k)[m]}, \tag{3.33}$$

for each annotator k, and

$$\text{mode}(t_i) = \arg\max_{j} \hat{p}\left(t_i = j|\text{MB}\left(t_i\right)\right) \tag{3.34}$$

for each document i.

3.4.4 Inference using Variational Bayesian Methods

Although Gibbs' sampling provides exact inference in the limit, when the number of annotators and/or the number of documents is very large (as we will see in Chapter 4), using Gibbs sampling for inference is no longer appropriate because of its low rate of convergence [Minka et al., 2014]. In this section, we present an alternative inference algorithm which provides fast approximate inference using variational Bayesian methods (or variational methods for short). The derivation of the variational equations for IBCC will be used in Chapter 4 where we introduce our contribution building on IBCC. We first describe the principle behind variational inference in Section 3.4.4.1. We then derive the specific variational equations for IBCC in Section 3.4.4.2.

3.4.4.1 Variational Bayesian Inference

The main idea behind variational methods is to approximate the posterior distribution in Equation 3.21 (omitting the hyperparameters for clarity) with a distribution

$$q\left(\mathbf{H}\right) \approx p\left(\mathbf{H}|\mathbf{V}\right),$$

that is computationally easier to work with. To find $q\left(\mathbf{H}\right)$, we first note the following.

Proposition 3.5. *For any choice of distribution $q\left(\mathbf{H}\right)$, the logarithm of the model evidence can be written*

$$lnp\left(\mathbf{V}\right) = \mathcal{L}\left(q\right) + KL\left(q||p\right) \tag{3.35}$$

where the first term is given by

$$\mathcal{L}\left(q\right) = E^{q}\left[ln\frac{p\left(\mathbf{X}\right)}{q\left(\mathbf{H}\right)}\right] \tag{3.36}$$

and the second term is the KL-divergence given by

$$KL\left(q||p\right)=-E^{q}\left[ln\frac{p\left(\mathbf{H}|\mathbf{V}\right)}{q\left(\mathbf{H}\right)}\right].$$

The expectations are taken with respect to the distribution q, that is $E^{q}\left[.\right]=\sum_{\mathbf{H}}.q\left(\mathbf{H}\right)$.

Proof. Assuming that Equation 3.35 is true, we have

$$\mathcal{L}\left(q\right)+\mathrm{KL}\left(q||p\right)=E^{q}\left[\ln\frac{p\left(\mathbf{X}\right)}{q\left(\mathbf{H}\right)}\right]-E^{q}\left[\ln\frac{p\left(\mathbf{H}|\mathbf{V}\right)}{q\left(\mathbf{H}\right)}\right].$$

Using the linearity property of the expectation, and the quotient rule of the logarithm, the right hand side gives

$$E^{q}\left[\ln\left(\frac{p\left(\mathbf{X}\right)}{\cancel{q\left(\mathbf{H}\right)}}.\frac{\cancel{q\left(\mathbf{H}\right)}}{p\left(\mathbf{H}|\mathbf{V}\right)}\right)\right].$$

Furthermore, multiplying both the numerator and denominator in the logarithm by $p\left(\mathbf{V}\right)$ gives

$$E^{q}\left[\ln\left(\frac{\cancel{p\left(\mathbf{X}\right)}p\left(\mathbf{V}\right)}{\cancel{p\left(\mathbf{H},\mathbf{V}\right)}}\right)\right]=E^{q}\left[\ln p\left(\mathbf{V}\right)\right].$$

Since $p\left(\mathbf{V}\right)$ is a constant, we get

$$\ln p\left(\mathbf{V}\right).E^{q}\left[1\right]=\ln p\left(\mathbf{V}\right)$$

which completes the proof. □

Since the KL-divergence satisfies $\mathrm{KL}\left(q||p\right)\geq 0$ (a result known as Gibbs' inequality) it follows from Equation 3.35 that the quantity $\mathcal{L}\left(q\right)$ forms the lower bound of $\ln p\left(\mathbf{V}\right)$ (since $-\infty\leq\ln p\left(\mathbf{V}\right)\leq 0$). As $q\left(\mathbf{H}\right)$ approaches $p\left(\mathbf{H}|\mathbf{V}\right)$, the KL-divergence vanishes to zero and the lower bound $\mathcal{L}\left(q\right)$ (also called the evidence lower bound, or ELBO) is maximised. Therefore, our goal is to find $q\left(\mathbf{H}\right)$ that minimises the KL-divergence $\mathrm{KL}\left(q||p\right)$, or equivalently, maximises the lower bound $\mathcal{L}\left(q\right)$. However, if we allow any possible choice for $q\left(\mathbf{H}\right)$, it follows that the maximum of the lower bound $\mathcal{L}\left(q\right)$ occurs when $q\left(\mathbf{H}\right)=p\left(\mathbf{H}\mid\mathbf{V}\right)$ which is intractable (that's why we are using variational approximation to begin with). We consider instead a restricted family of distributions $q\left(\mathbf{H}\right)$, and then seek the member of this family for which the KL divergence is minimised. One way to restrict the family of approximating distributions is to use a parametric distribution $q\left(\mathbf{H};\boldsymbol{\theta}\right)$ governed by a set of variational parameters $\boldsymbol{\theta}$. The lower bound $\mathcal{L}\left(q\right)$ then becomes a function of $\boldsymbol{\theta}$, and we can exploit standard nonlinear optimisation techniques to determine the optimal values for the parameters. Alternatively, a common restriction is to assume that $q\left(\mathbf{H}\right)$ factorises over a partition of $\mathbf{H}$ of disjoint groups of variables $\mathbf{H}_{i}$, such that

$$q\left(\mathbf{H}\right)=\prod_{i}q\left(\mathbf{H}_{i}\right).\tag{3.37}$$

Note that for the observed nodes $\mathbf{V}$, there is no factor $q\left(V_i\right)$ in the variational distribution $q\left(\mathbf{H}\right)$. Given the functional form of $q\left(\mathbf{H}\right)$ in Equation 3.37, results from the mean field theory [Parisi, 1988] show that the optimal solution for each factor is

$$q^*\left(\mathbf{H}_i\right) = \frac{1}{Z}\exp\left(E_{\overline{\mathbf{H}_i}}\left[\ln p\left(\mathbf{X}\right)\right]\right) \tag{3.38}$$

where

$$Z = \sum_{\mathbf{H}_i}\exp\left(E_{\overline{\mathbf{H}_i}}\left[\ln p\left(\mathbf{X}\right)\right]\right) \tag{3.39}$$

is a normalisation constant, the set $\overline{\mathbf{H}_i}$ is defined such that $\overline{\mathbf{H}_i} = \mathbf{H}\backslash\mathbf{H}_i$, and $E_{\overline{\mathbf{H}_i}}\left[.\right] = \sum_{\overline{\mathbf{H}_i}} .q\left(\overline{\mathbf{H}_i}\right)$ is the expectation taken with respect to all factors except $\mathbf{H}_i$. Note that the solution $q^*\left(\mathbf{H}_i\right)$ (Equation 3.38) is coupled since it depends on the expectation with respect to the other factors $\overline{\mathbf{H}_i}$. Therefore, mean-field variational optimisation proceeds by initialising each factor $q\left(\mathbf{H}_i\right)$ and cycling through them in turn, replacing the current distribution with a revised estimate given by Equation 3.38. Unlike Gibbs' sampling, each iteration is guaranteed to decrease the divergence $\mathrm{KL}\left(q||p\right)$ and converge to a local minimum in a similar fashion to standard expectation maximisation algorithms [Simpson, 2014]. Once the factors $q^*\left(\mathbf{H}_i\right)$ have converged, we can approximate the value of the hidden variables $\mathbf{H}$ by computing their expected values.

3.4.4.2 Mean-field Variational Equations for IBCC

Following Simpson et al. [2013] we derive the optimal variational equations as applied to IBCC. We propose to factorise the factors from Equation 3.37 by grouping the latent variables representing the parameters of IBCC as follows

$$q\left(\mathbf{t},\boldsymbol{\kappa},\boldsymbol{\Pi}\right) = q\left(\mathbf{t}\right)q\left(\boldsymbol{\kappa},\boldsymbol{\Pi}\right). \tag{3.40}$$

From Equation 3.38, the optimal factor for $\mathbf{t}$ is given by

$$q^*\left(\mathbf{t}\right) = \frac{1}{Z}\exp\left(E_{\boldsymbol{\kappa},\boldsymbol{\Pi}}\left[\ln p\left(\mathbf{t},\mathbf{c},\boldsymbol{\kappa},\boldsymbol{\Pi};\boldsymbol{\nu},\mathbf{A}\right)\right]\right).$$

Using Equation 3.17, we get

$$\begin{aligned} q^*\left(\mathbf{t}\right) \quad = \quad & \frac{1}{Z}\exp\left(E_{\boldsymbol{\kappa},\boldsymbol{\Pi}}\left[\ln p\left(\mathbf{t},\mathbf{c}|\boldsymbol{\kappa},\boldsymbol{\Pi}\right)\right]\right) \times \\ & \exp\left(E_{\boldsymbol{\kappa}}\left[\ln p\left(\boldsymbol{\kappa};\boldsymbol{\nu}\right)\right] + E_{\boldsymbol{\Pi}}\left[\ln p\left(\boldsymbol{\Pi};\mathbf{A}\right)\right]\right) \end{aligned}$$

where the normalising constant Z is given according to Equation 3.39

$$\begin{aligned}
Z &= \sum_{\mathbf{t}} \exp\left(E_{\boldsymbol{\kappa},\boldsymbol{\Pi}}\left[\ln\left(p\left(\mathbf{t},\mathbf{c}|\boldsymbol{\kappa},\boldsymbol{\Pi}\right)p\left(\boldsymbol{\kappa};\boldsymbol{\nu}\right)p\left(\boldsymbol{\Pi};\mathbf{A}\right)\right)\right]\right) \\
&= \exp\left(E_{\boldsymbol{\kappa}}\left[\ln p\left(\boldsymbol{\kappa};\boldsymbol{\nu}\right)\right] + E_{\boldsymbol{\Pi}}\left[\ln p\left(\boldsymbol{\Pi};\mathbf{A}\right)\right]\right) \times \\
&\quad \sum_{\mathbf{t}} \exp\left(E_{\boldsymbol{\kappa},\boldsymbol{\Pi}}\left[\ln p\left(\mathbf{t},\mathbf{c}|\boldsymbol{\kappa},\boldsymbol{\Pi}\right)\right]\right).
\end{aligned}$$

Combining the terms and simplying the exponential terms gives

$$q^*\left(\mathbf{t}\right) = \frac{\exp\left(E_{\boldsymbol{\kappa},\boldsymbol{\Pi}}\left[\ln p\left(\mathbf{t},\mathbf{c}|\boldsymbol{\kappa},\boldsymbol{\Pi}\right)\right]\right)}{\sum_{\mathbf{t}} \exp\left(E_{\boldsymbol{\kappa},\boldsymbol{\Pi}}\left[\ln p\left(\mathbf{t},\mathbf{c}|\boldsymbol{\kappa},\boldsymbol{\Pi}\right)\right]\right)}. \tag{3.41}$$

Since the target topics t_i are independent, we are looking for the optimal factor $q^*\left(t_i = j\right)$ of each target topic such that

$$q^*\left(\mathbf{t}\right) = \prod_{i=1}^{I} q^*\left(t_i\right).$$

Since each t_i is distributed according to a categorical distribution, this gives

$$q^*\left(\mathbf{t}\right) = \prod_{i=1}^{I}\prod_{j=1}^{J} q^*\left(t_i = j\right)^{\delta\left(t_i - j\right)}.$$

To find the optimal terms $q^*\left(t_i\right)$, we first note that the terms in Equation 3.41 can be written (using Equation 3.19)

$$\exp\left(E_{\boldsymbol{\kappa},\boldsymbol{\Pi}}\left[\ln p\left(\mathbf{t},\mathbf{c}|\boldsymbol{\kappa},\boldsymbol{\Pi}\right)\right]\right) = \prod_{i=1}^{I}\exp\left(E_{\boldsymbol{\kappa}}\left[\ln\kappa_{t_i}\right] + \sum_{k=1}^{K} E_{\boldsymbol{\Pi}}\left[\ln\pi^{(k)}_{t_i,c_i^{(k)}}\right]\right).$$

Defining

$$\rho_{i,j} = \exp\left(E_{\boldsymbol{\kappa}}\left[\ln\kappa_j\right] + \sum_{k=1}^{K} E_{\boldsymbol{\Pi}}\left[\ln\pi^{(k)}_{j,c_i^{(k)}}\right]\right), \tag{3.42}$$

the optimal factor $q^*\left(t_i = j\right)$ is given by

$$q^*\left(t_i = j\right) = \frac{\rho_{i,j}}{\sum_{\iota=1}^{J}\rho_{\iota,j}}. \tag{3.43}$$

Since $q^*\left(t_i\right)$ is a categorical distribution, we have

$$E_{\mathbf{t}}\left[t_i = j\right] = q^*\left(t_i = j\right). \tag{3.44}$$

To simplify subsequent equations, we introduce the following definitions of the expectation of the number of occurrences of each topic in $\mathbf{t}$

$$N_j = \sum_{i=1}^{I} E_{\mathbf{t}}\left[t_i = j\right], \tag{3.45}$$

and the expectation of the number of occurrence of each judgment $c_i^{(k)} = l$ when the topic was $t_i = j$

$$M_{j,l} = \sum_{i=1}^{I} \delta\left(c_i^{(k)} - l\right) E_{\mathbf{t}}\left[t_i = j\right]. \tag{3.46}$$

For the model's parameters, the optimal factor is given by

$$q^*\left(\boldsymbol{\kappa}, \boldsymbol{\Pi}\right) = \frac{1}{Z}\exp\left(E_{\mathbf{t}}\left[\ln p\left(\mathbf{t}, \mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}; \boldsymbol{\nu}, \mathbf{A}\right)\right]\right).$$

Using Equation 3.17, the expectation is given by

$$E_{\mathbf{t}}\left[\ln p\left(\mathbf{t}, \mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}; \boldsymbol{\nu}, \mathbf{A}\right)\right] \quad = \quad E_{\mathbf{t}}\left[\ln p\left(\mathbf{t}, \mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}\right)\right] + \ln p\left(\boldsymbol{\kappa}; \boldsymbol{\nu}\right) + \ln p\left(\boldsymbol{\Pi}; \mathbf{A}\right)$$

Using Equation 3.19, the first term gives

$$E_{\mathbf{t}}\left[\ln p\left(\mathbf{t}, \mathbf{c} | \boldsymbol{\kappa}, \boldsymbol{\Pi}\right)\right] = E_{\mathbf{t}}\left[\sum_{i=1}^{I} \ln\left(\kappa_{t_i}\right)\right] + E_{\mathbf{t}}\left[\sum_{i=1}^{I}\sum_{k=1}^{K} \ln\left(\pi^{(k)}_{t_i, c_i^{(k)}}\right)\right].$$

Combining the terms and grouping the ones independent of $\boldsymbol{\kappa}$ and $\boldsymbol{\Pi}$ gives

$$\begin{aligned} E_{\mathbf{t}}\left[\ln p\left(\mathbf{t}, \mathbf{c}, \boldsymbol{\kappa}, \boldsymbol{\Pi}; \boldsymbol{\nu}, \mathbf{A}\right)\right] \quad = \quad & \left(E_{\mathbf{t}}\left[\sum_{i=1}^{I} \ln\left(\kappa_{t_i}\right)\right] + \ln p\left(\boldsymbol{\kappa}; \boldsymbol{\nu}\right)\right) + \\ & \left(E_{\mathbf{t}}\left[\sum_{i=1}^{I}\sum_{k=1}^{K} \ln\left(\pi^{(k)}_{t_i, c_i^{(k)}}\right)\right] + \ln p\left(\boldsymbol{\Pi}; \mathbf{A}\right)\right). \end{aligned}$$

Therefore, we observe that the joint distribution of $q^*\left(\boldsymbol{\kappa}, \boldsymbol{\Pi}\right)$ factorises over each parameter $\boldsymbol{\kappa}$ and $\boldsymbol{\Pi}$ such that

$$q^*\left(\boldsymbol{\kappa}, \boldsymbol{\Pi}\right) = q^*\left(\boldsymbol{\kappa}\right) q^*\left(\boldsymbol{\Pi}\right).$$

It follows that the optimal factor for $\boldsymbol{\kappa}$ is given by

$$q^*\left(\boldsymbol{\kappa}\right) = \frac{1}{Z}\exp\left(E_{\mathbf{t}}\left[\sum_{i=1}^{I} \ln\left(\kappa_{t_i}\right)\right] + \ln p\left(\boldsymbol{\kappa}; \boldsymbol{\nu}\right)\right). \tag{3.47}$$

Since $p\left(\boldsymbol{\kappa} | \boldsymbol{\nu}\right)$ is a Dirichlet distribution, its logarithm is given by Equation A.2 (Appendix A) leading to

$$\ln p\left(\boldsymbol{\kappa}; \boldsymbol{\nu}\right) = \sum_{j=1}^{J} \left(\nu_j - 1\right) \ln\kappa_j - \ln\mathrm{B}\left(\boldsymbol{\kappa}\right).$$

Furthermore, making use of Equation 3.27, we get

$$E_{\mathbf{t}}\left[\sum_{i=1}^{I} \ln\kappa_{t_i}\right] = E_{\mathbf{t}}\left[\ln\prod_{i=1}^{I} \kappa_{t_i}\right] = E_{\mathbf{t}}\left[\ln\prod_{j=1}^{J} \kappa_j^{N_j}\right] = \sum_{j=1}^{J} N_j \ln\kappa_j.$$

Substituting both terms into Equation 3.47, we get

$$
\begin{aligned}
q^{*}(\boldsymbol{\kappa}) &= \frac{1}{Z}\mathrm{exp}\left(\sum_{j=1}^{J} N_j \mathrm{ln}\kappa_j + \sum_{j=1}^{J} (\nu_j - 1)\,\mathrm{ln}\kappa_j - \mathrm{lnB}(\boldsymbol{\kappa})\right) \\
&= \frac{1}{Z}\mathrm{exp}\left(\sum_{j=1}^{J} (N_j + \nu_j - 1)\,\mathrm{ln}\kappa_j\right)\mathrm{exp}\left(-\mathrm{lnB}(\boldsymbol{\kappa})\right) \\
&\propto \mathrm{exp}\left(\mathrm{ln}\prod_{j=1}^{J} \kappa_j^{N_j+\nu_j-1}\right) \\
&= \mathrm{Dir}(\boldsymbol{\kappa}|\mathbf{N}+\boldsymbol{\nu}). \qquad (3.48)
\end{aligned}
$$

Since $q^{*}(\boldsymbol{\kappa})$ is a Dirichlet distribution, the expectation required in Equation 3.42 is given by

$$
E_{\boldsymbol{\kappa}}[\mathrm{ln}\kappa_j] = \psi(\nu_j + N_j) - \psi\left(\sum_{\iota=1}^{J} \nu_\iota + N_\iota\right) \qquad (3.49)
$$

where $\psi(.)$ is the digamma function $\psi(x) = \frac{d}{dx}\mathrm{ln}(\Gamma(x)) = \frac{\Gamma'(x)}{\Gamma(x)}$.Similarly, the optimal factor for $\mathbf{\Pi}$ is given by

$$
q^{*}(\mathbf{\Pi}) = \frac{1}{Z}\mathrm{exp}\left(E_{\mathbf{t}}\left[\sum_{i=1}^{I}\sum_{k=1}^{K} \mathrm{ln}\pi_{t_i,c_i^{(k)}}^{(k)}\right] + \prod_{k=1}^{K}\prod_{j=1}^{J} \mathrm{ln}p\left(\boldsymbol{\pi}_j^{(k)};\boldsymbol{\alpha}_j^{(k)}\right)\right).
$$

The factor $q^{*}(\mathbf{\Pi})$ can be factorise such that

$$
q^{*}(\mathbf{\Pi}) = \prod_{k=1}^{K}\prod_{j=1}^{J} q^{*}\left(\boldsymbol{\pi}_j^{(k)}\right),
$$

where each factor is given by

$$
q^{*}\left(\boldsymbol{\pi}_j^{(k)}\right) = \frac{1}{Z}\mathrm{exp}\left(E_{\mathbf{t}}\left[\sum_{i=1}^{I} \mathrm{ln}\pi_{t_i,c_i^{(k)}}^{(k)}\right] + \mathrm{ln}p\left(\boldsymbol{\pi}_j^{(k)};\boldsymbol{\alpha}_j^{(k)}\right)\right).
$$

Since $p\left(\boldsymbol{\pi}_j^{(k)}|\boldsymbol{\alpha}_j^{(k)}\right)$ is a Dirichlet distribution (Equation 3.16), Equation A.2 (Appendix A) gives the logarithm of $\boldsymbol{\pi}_j^{(k)}$ as follow

$$
\mathrm{ln}p\left(\boldsymbol{\pi}_j^{(k)};\boldsymbol{\alpha}_j^{(k)}\right) = \sum_{l=1}^{L}\left(\boldsymbol{\alpha}_{j,l}^{(k)} - 1\right)\mathrm{ln}\pi_{j,l}^{(k)} - \mathrm{lnB}\left(\boldsymbol{\pi}_j^{(k)}\right).
$$

Furthermore, making use of Equation 3.30, we get

$$
E_{\mathbf{t}}\left[\sum_{i=1}^{I} \mathrm{ln}\pi_{t_i,c_i^{(k)}}^{(k)}\right] = E_{\mathbf{t}}\left[\mathrm{ln}\prod_{i=1}^{I} \pi_{t_i,c_i^{(k)}}^{(k)}\right] = E_{\mathbf{t}}\left[\mathrm{ln}\prod_{l=1}^{L}\left(\pi_{j,l}^{(k)}\right)^{M_{j,l}^{(k)}}\right] = \sum_{l=1}^{L} M_{j,l}^{(k)}\mathrm{ln}\pi_{j,l}^{(k)}.
$$

Substituting, we get

$$\begin{aligned}
q^*\left(\boldsymbol{\pi}_j^{(k)}\right) &= \frac{1}{Z}\exp\left(\sum_{l=1}^{L} M_{j,l}^{(k)}\ln\pi_{j,l}^{(k)} + \sum_{l=1}^{L}\left(\boldsymbol{\alpha}_{j,l}^{(k)}-1\right)\ln\pi_{j,l}^{(k)} - \ln\mathrm{B}\left(\boldsymbol{\pi}_j^{(k)}\right)\right) \\
&= \frac{1}{Z}\exp\left(\sum_{l=1}^{L}\left(M_{j,l}^{(k)}+\boldsymbol{\alpha}_{j,l}^{(k)}-1\right)\ln\pi_{j,l}^{(k)}\right)\exp\left(-\ln\mathrm{B}\left(\boldsymbol{\pi}_j^{(k)}\right)\right) \\
&\propto \exp\left(\ln\prod_{l=1}^{L}\left(\pi_{j,l}^{(k)}\right)^{M_{j,l}^{(k)}+\boldsymbol{\alpha}_{j,l}^{(k)}-1}\right) \\
&= \mathrm{Dir}\left(\boldsymbol{\pi}_j^{(k)}|\mathbf{M}_j^{(k)}+\boldsymbol{\alpha}^{(k)}\right).
\end{aligned} \tag{3.50}$$

The expectation required in Equation 3.42 is given by

$$E_{\boldsymbol{\Pi}}\left[\ln\pi_{j,l}^{(k)}\right] = \psi\left(\alpha_{j,l}^{(k)}+M_{j,l}^{(k)}\right) - \psi\left(\sum_{l=1}^{L}\alpha_{j,l}^{(k)}+M_{j,l}^{(k)}\right). \tag{3.51}$$

The variational inference algorithm as applied to IBCC (Algorithm 4) iterates between updating the latent variable $\mathbf{t}$ and the parameters $\{\boldsymbol{\kappa}, \boldsymbol{\Pi}\}$, taking as input the annotators' judgments $\mathbf{c}$ (Algorithm 4).

Algorithm 4 Variational inference for IBCC.

```
Inputs: nb iterations $M > 0$, $\boldsymbol{\nu}$, $\mathbf{A}$, $\mathbf{c}$

Initialise all $E_{\boldsymbol{\Pi}}\left[\ln\pi_{j,l}^{(k)}\right]^{[0]}$ and $E_{\boldsymbol{\kappa}}\left[\ln\kappa_j\right]^{[0]}$

for each iteration $m$ in $\{1, \cdots, M\}$ do
    for each document $i$ in $\{1, \cdots, I\}$ do                    ▷ E-step
        Compute $E_{\mathbf{t}}\left[t_i = j\right]^{[m]}$
    end for

    for each topic $j$ in $\{1, \cdots, J\}$ do                       ▷ M-step
        Compute $E_{\boldsymbol{\kappa}}\left[\ln\kappa_j\right]^{[m]}$
        for each annotator $k$ in $\{1, \cdots, K\}$ do
            for each topic $l$ in $\{1, \cdots, L\}$ do
                Compute $E_{\boldsymbol{\Pi}}\left[\ln\pi_{j,l}^{(k)}\right]^{[m]}$
            end for
        end for
    end for
end for

return $q^*\left(\mathbf{t}, \boldsymbol{\kappa}, \boldsymbol{\Pi}\right)$
```

Once convergence is observed, we can compute the optimal variational solution from Equation 3.43, Equation 3.48, and Equation 3.50, and take their expectation as the inferred point value, that is

$$E_{\mathbf{t}}\left[t_i = j\right] = q^*\left(t_i = j\right), \tag{3.52}$$

$$E_{\boldsymbol{\kappa}}\left[\kappa_j\right] = \frac{\mathbf{N}_j + \boldsymbol{\nu}_j}{\sum_{\iota=1}^{J} \mathbf{N}_\iota + \boldsymbol{\nu}_\iota}, \tag{3.53}$$

and

$$E_{\mathbf{\Pi}}\left[\pi_{j,l}^{(k)}\right] = \frac{\mathbf{M}_l^{(k)} + \boldsymbol{\alpha}_l^{(k)}}{\sum_{\iota=1}^{J} \mathbf{M}_{j,\iota}^{(k)} + \boldsymbol{\alpha}_\iota^{(k)}}. \tag{3.54}$$

Our discussion so far assumed that each document has a single target topic t_i that is to be inferred. Indeed, this is one of the main assumptions of IBCC. However, as per Aim 1, we wish to aggregate judgments from annotators for documents with multiple topics. In the next section, we describe the generative process handling documents with multiple topics.

3.4.5 Generative Process for Documents with Multiple Topics

In this section, we extend Algorithm 1 (i.e. the generative process for documents with single topics) to also handle the case when documents have multiple topics. The new generative process to obtain the judgements $\mathbf{c}$ for documents with multiple topics is described in Algorithm 5. In particular, we assign a topic distribution $\mathbf{\Lambda}_i$ of dimension J (one for each alternative) with each document i instead of the single topic proportion $\boldsymbol{\kappa}$ shared across all documents (note the change in Line 4 of Algorithm 1 from $\boldsymbol{\kappa}$ to $\mathbf{\Lambda}_i$ in Line 4 of Algorithm 5). In contrast with Algorithm 1, each annotator is now being asked to judge a document with target topic z_i drawn at random from the topic proportion $\mathbf{\Lambda}_i$ (line 5 in Algorithm 5), such that,

$$z_i \sim \text{Cat}\left(\Lambda_i\right),$$

instead of the unique target topic t_i drawn from $\boldsymbol{\kappa}$ (Line 4 in Algorithm 1). The judgment $c_i^{(k)}$ for each annotator k is a sample from his confusion matrix as before, but conditioned on the value of the sample z_i ($\boldsymbol{\pi}_{z_i}^{(k)}$ in Line 6 of Algorithm 5, instead of $\boldsymbol{\pi}_{t_i}^{(k)}$ in Line 6 of Algorithm 1).

Algorithm 5 Generative process for documents with multiple topics.

```
Input: the topic proportions $\mathbf{\Lambda}$ and the confusion matrices $\mathbf{\Pi}^{(k)}$

for each document $i \in \{1, \cdots, I\}$ do                          ▷ for each document $i$
    Sample $z_i \sim \text{Categorical}(\mathbf{\Lambda}_i)$                  ▷ sample a target topic
    for each annotator $k \in \{1, \cdots, K\}$ do                   ▷ for each annotator $k$
        Sample $c_i^{(k)} \sim \text{Categorical}\left(\boldsymbol{\pi}_{z_i}^{(k)}\right)$   ▷ sample the annotator's judgment
    end for
end for

return c
```

In the special case where the documents have a single topic t_i, the distribution $\mathbf{\Lambda}_i$ is a *point mass* (i.e. a degenerate distribution) [Karr, 1993, p. 31], such that,

$$\Lambda_{i,j} = \begin{cases} 1 & \text{if } j = t_i, \\ 0 & \text{otherwise.} \end{cases} \tag{3.55}$$

From Equation 3.55, it follows that

$$\text{mode}\,(\mathbf{\Lambda}_i) = t_i. \tag{3.56}$$

In such cases where documents have a single topic, Algorithm 5 is equivalent to Algorithm 1 since the sample z_i is equal to t_i with probability one. This new generative process (Algorithm 5) is used to generate the synthetic dataset in our empirical evaluation of IBCC in the Section 3.4.7.

3.4.6 Worked Example

To make the description of IBCC concrete, we look at a simple example in which the goal is to demonstrate the mechanism by which IBCC infers the target topics of a set of documents based on the annotators' judgment. The exemplar documents are a corpus of ten news articles which have been analysed by a topic model. It is important to note that a minimum number of documents are necessary to correctly infer the target topics (the exact number depends on the number and accuracy of the annotators, and the proportion of documents of each topic in the corpus). We assume that each topic is labelled with the highest-probability word within that topic. For example, the topic ['Markets', 'buyers', 'sellers', ...] is labelled 'Markets'. In this example, we will particularly pay attention to the first document which will serve as a representative baseline. To simplify the example further, we use a subset of $J = 2$ topics (namely topic 1:“Markets” and topic 2:“Companies”)[1].

3.4.6.1 Inference for Documents with a Single Topic

The target topic associated with each document in the corpus is as follows,

$$\mathbf{t} = \begin{bmatrix} 1 & 1 & 2 & 1 & 2 & 2 & 1 & 2 & 1 & 2 \end{bmatrix}.$$

That is, the target topic t_1 for the first document is 1:“Markets”, and the target topic t_3 for the third document is 2:“Companies”. We consider a set of $K = 3$ annotators providing a single judgment regarding each of the ten documents in the corpus[2]. In particular, the first annotator provides the following judgment

$$\mathbf{c}^{(1)} = \begin{bmatrix} \textcircled{2} & 1 & 2 & 1 & 2 & \textcircled{1} & 1 & 2 & 1 & 2 \end{bmatrix}.$$

The second annotator provides

$$\mathbf{c}^{(2)} = \begin{bmatrix} 1 & 1 & \textcircled{1} & 1 & \textcircled{1} & \textcircled{1} & 1 & \textcircled{1} & 1 & \textcircled{1} \end{bmatrix}.$$

Finally, the third annotator gives

$$\mathbf{c}^{(3)} = \begin{bmatrix} 1 & 1 & 2 & 1 & 2 & 2 & 1 & 2 & 1 & 2 \end{bmatrix}.$$

Since the number of documents and annotators is small, we use Gibbs' sampling as described in Section 3.4.3 to obtain an exact inference of the target topics $\hat{\mathbf{t}}$. Given the above annotators' judgments $\mathbf{c}^{(k)}$, we step through Algorithm 3 and withhold the ground truth $\mathbf{t}$ from the algorithm for later comparison with the inferred value $\hat{\mathbf{t}}$.

Line 1 We arbitrarily set the number of samples of the Markov chain to $M = 500$. This value is enough for the algorithm to converge given the small amount of documents in the corpus, the small number of topics and the small number of annotators. Furthermore, we set the same uninformative hyperparameter to the prior distribution over the confusion matrix of each annotator to reflect our uncertainty over their accuracy and bias, that is,

$$\boldsymbol{\alpha}_j^{(k)} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

for all topics $j \in \{1, 2\}$, or equivalently,

$$\mathbf{A}^{(k)} = \begin{bmatrix} \boldsymbol{\alpha}_0^{(k)} \\ \boldsymbol{\alpha}_1^{(k)} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

[1] We selected these two topics because they are both relevant to the first document but in different proportions.

[2] The judgments flagged with a circle are those which disagree with the ground truth.

for each annotator $k \in \{1,2,3\}$. We also set an uninformative hyperparameter to the prior distribution over the topic proportion, that is,

$$\boldsymbol{\nu} = \left[\begin{array}{cc} 1 & 1 \end{array}\right],$$

to reflect our uncertainty over the proportion of each topic in the corpus.

Line 2 Next, we initialise the Markov chain of the Gibbs' sampling algorithm. Since the initial values affect its rate of convergence, different strategies may be used (e.g. random assignment or expectation maximisation). In this example, we randomly assign the initial values of the Markov chain for the confusion matrices according to a uniform distribution, giving,

$$\mathbf{\Pi}^{(1)[0]} = \begin{bmatrix} 0.47 & 0.53 \\ 0.72 & 0.28 \end{bmatrix}, \quad \mathbf{\Pi}^{(2)[0]} = \begin{bmatrix} 0.13 & 0.87 \\ 0.77 & 0.23 \end{bmatrix}, \quad \mathbf{\Pi}^{(3)[0]} = \begin{bmatrix} 0.51 & 0.49 \\ 0.87 & 0.13 \end{bmatrix}$$

for annotators $k = \{1,2,3\}$ respectively. We proceed similarly with the topic proportion,

$$\boldsymbol{\kappa}^{[0]} = \left[\begin{array}{cc} 0.23 & 0.77 \end{array}\right]$$

and the topic assignment to each document,

$$\mathbf{t}^{[0]} = \left[\begin{array}{cccccccccc} 1 & 2 & 1 & 1 & 2 & 1 & 1 & 2 & 1 & 1 \end{array}\right].$$

Line 5 We draw a sample for the topic proportion $\boldsymbol{\kappa}^{[1]}$ from Equation 3.25 according to

$$\boldsymbol{\kappa}^{[1]}|\mathbf{t}^{[0]}, \boldsymbol{\nu} \sim \mathrm{Dir}\left(\boldsymbol{\nu} + \boldsymbol{N}^{[0]}\right) = \mathrm{Dir}\left(\left[\begin{array}{cc} 8 & 4 \end{array}\right]\right),$$

where $\boldsymbol{N}^{[0]} = \left[\begin{array}{cc} 7 & 3 \end{array}\right]$ is the number of topics j in $\mathbf{t}^{[0]}$. This gives

$$\boldsymbol{\kappa}^{[1]} = \left[\begin{array}{cc} 0.92 & 0.08 \end{array}\right].$$

Line 9 In order to sample a confusion matrix $\mathbf{\Pi}^{(k)[1]}$ for each annotator from Equation 3.28, we first need to compute the matrix of count $\mathbf{M}^{(k)[0]}$ associated with each annotator. To illustrate, the count $\mathbf{M}^{(3)[0]}_{1,2}$ is the number of times the third annotator provided the judgments 2:"Companies" for documents where the sampled topic assignment in $\mathbf{t}^{[0]}$ was 1:"Markets". In this case, the value of $\mathbf{M}^{(3)[0]}_{1,2}$ is 3. Proceeding as follows for each annotator, we obtain

$$\mathbf{M}^{(1)[0]} = \begin{bmatrix} 4 & 3 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{M}^{(2)[0]} = \begin{bmatrix} 7 & 0 \\ 3 & 0 \end{bmatrix}, \quad \mathbf{M}^{(3)[0]} = \begin{bmatrix} 4 & 3 \\ 1 & 2 \end{bmatrix}.$$

Next, for each annotator $k \in \{1,2,3\}$ and each topic $j \in \{1,2\}$, we sample a row of the confusion matrix $\boldsymbol{\pi}^{(k)[1]}_j$ from Equation 3.28. The first row $j = 1$ of the first annotator

$k = 1$ is given by

$$\boldsymbol{\pi}_1^{(1)[1]} | \mathbf{t}^{[0]}, \mathbf{c}, \boldsymbol{\alpha}_1^{(1)} \sim \text{Dir}\left(\boldsymbol{\alpha}_{0,1}^{(1)} + \mathbf{M}_1^{(1)[0]}\right) = \text{Dir}\left(\left[\begin{array}{cc} 5 & 4 \end{array}\right]\right),$$

where $\boldsymbol{\alpha}_1^{(1)} = \left[\begin{array}{cc} 1 & 1 \end{array}\right]$ (i.e. the first row of the hyperparameter $\mathbf{A}^{(1)}$ of the first annotator) and $\mathbf{M}_1^{(1)[0]} = \left[\begin{array}{cc} 4 & 3 \end{array}\right]$ (i.e. the first row of the count matrix $\mathbf{M}^{(1)[0]}$ of the first annotator). The resulting sample $\boldsymbol{\pi}_1^{(1)[1]}$ is

$$\boldsymbol{\pi}_1^{(1)[1]} = \left[\begin{array}{cc} 0.51 & 0.49 \end{array}\right].$$

The same step is repeated for each row of each annotators' confusion matrix, giving

$$\boldsymbol{\Pi}^{(1)[1]} = \left[\begin{array}{cc} 0.51 & 0.49 \\ 0.63 & 0.37 \end{array}\right], \quad \boldsymbol{\Pi}^{(2)[1]} = \left[\begin{array}{cc} 1 & 0 \\ 1 & 0 \end{array}\right], \quad \boldsymbol{\Pi}^{(3)[1]} = \left[\begin{array}{cc} 0.64 & 0.36 \\ 0.5 & 0.55 \end{array}\right].$$

At this point we start seeing a convergence of the confusion matrices towards a more accurate assessment of the annotators' accuracy and bias as compared to the random samples in $\boldsymbol{\Pi}^{(k)[0]}$ in Line 2.

Line 15 For each document i in the corpus and each topic j, Equation 3.31 gives the probability of the topic j being the target topic of document i. For the first document, the probability of this document to be of target topic 1:"Markets" is given by,

$$p_{1,1}^{[1]} = p\left(t_1^{[1]} = 1 | \boldsymbol{\kappa}^{[1]}, \boldsymbol{\pi}_1^{(k)[1]}, \mathbf{c}\right) = \frac{\kappa_1^{[1]} \prod_{k=1}^{3} \pi_{1,c_1^{(k)}}^{(k)[1]}}{\kappa_1^{[1]} \prod_{k=1}^{3} \pi_{1,c_1^{(k)}}^{(k)[1]} + \kappa_2^{[1]} \prod_{k=1}^{3} \pi_{2,c_1^{(k)}}^{(k)[1]}}.$$

The numerical value of the numerator is given by

$$\begin{aligned} \kappa_1^{[1]} \prod_{k=1}^{3} \pi_{1,c_1^{(k)}}^{(k)[1]} &= 0.92 \times (0.49 \times 1 \times 0.64) \\ &= 0.288, \end{aligned}$$

while the second term of the denominator is given by,

$$\begin{aligned} \kappa_2^{[1]} \prod_{k=1}^{3} \pi_{2,c_1^{(k)}}^{(k)[1]} &= 0.08 \times (0.37 \times 1 \times 0.5) \\ &= 0.0148. \end{aligned}$$

Combining the terms gives

$$p_{1,1}^{[1]} = 0.95.$$

Similarly, the probability of the first document being of topic 2:“Companies” is given by

$$p_{1,2}^{[1]} = p\left(t_1^{[1]} = 2|\boldsymbol{\kappa}^{[1]}, \boldsymbol{\pi}_2^{(k)[1]}, \mathbf{c}\right) = \frac{\kappa_2^{[1]} \prod_{k=1}^{3} \pi_{2,c_1^{(k)}}^{(k)[1]}}{\kappa_1^{[1]} \prod_{k=1}^{3} \pi_{1,c_1^{(k)}}^{(k)[1]} + \kappa_2^{[1]} \prod_{k=1}^{3} \pi_{2,c_1^{(k)}}^{(k)[1]}}.$$

Re-using and combining the two terms from our computations above gives

$$p_{1,2}^{[1]} = 0.05.$$

Line 17 The two values $p_{1,1}^{[1]}$ and $p_{1,2}^{[1]}$ together form the parameter $\mathbf{p}_1^{[1]}$ of the categorical distribution over topics of the first document for the first iteration of the Gibbs' sampling algorithm, that is

$$t_1^{[1]} \sim \text{Categorical}\left(\mathbf{p}_1^{[1]}\right) = \text{Categorical}\left(\left[\begin{array}{cc} 0.95 & 0.05 \end{array}\right]\right).$$

Drawing a sample from this distribution gives

$$t_1^{[1]} = 1,$$

meaning that the first sample for the topic of the first document is 1:“Markets”. We repeat this procedure for each of the nine remaining documents, giving

$$\mathbf{t}^{[1]} = \left[\begin{array}{cccccccccc} 1 & 1 & 2 & 1 & 2 & 2 & 1 & 2 & 1 & 1 \end{array}\right].$$

Taken together the value of $\left\{\boldsymbol{\kappa}^{[1]}, \boldsymbol{\Pi}^{[1]}, \mathbf{t}^{[1]}\right\}$ form a unique Gibbs' sample from the posterior distribution defined in Equation 3.21. This sampling is repeated $M-1$ times to obtain the empirical posterior distributions of each variable (Figure 3.5 for $\boldsymbol{\kappa}$, and Figure 3.6 for t_1).

Given the empirical posterior distribution of the topic proportion in Figure 3.5a and Figure 3.5b, we take the mean of $\boldsymbol{\kappa}$ according to Equation 3.32 as a point value estimate, that is

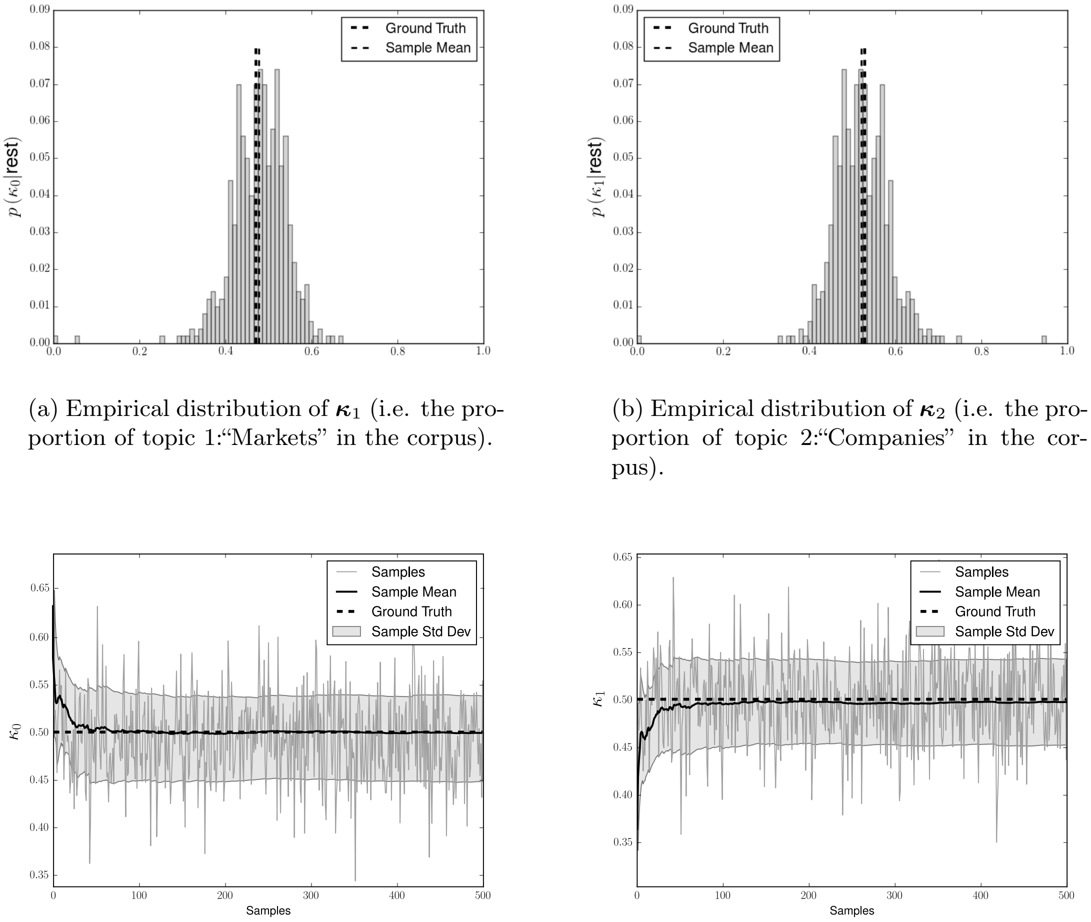
$$\overline{\boldsymbol{\kappa}} = \left[\begin{array}{cc} 0.5 & 0.5 \end{array}\right].$$

In this instance, the value of $\overline{\boldsymbol{\kappa}}$ is equal to the ground truth $\boldsymbol{\kappa}$ since $\mathbf{t}$ has an equal number of documents with target topic of each type. Furthermore, using the mode as a point estimate of $\mathbf{t}$ (Equation 3.34) on the empirical posterior distribution in Figure 3.6a gives

$$\text{mode}\left(t_1\right) = 1,$$

meaning that the inferred topic of the first document is 1:“Markets”. And similarly for the other documents in the corpus,

$$\text{mode}\left(\mathbf{t}\right) = \left[\begin{array}{cccccccccc} 1 & 1 & 2 & 1 & 2 & 2 & 1 & 2 & 1 & 2 \end{array}\right].$$



(a) Empirical distribution of $\boldsymbol{\kappa}_1$ (i.e. the proportion of topic 1:"Markets" in the corpus).

(b) Empirical distribution of $\boldsymbol{\kappa}_2$ (i.e. the proportion of topic 2:"Companies" in the corpus).

(c) Convergence of the mean of $\boldsymbol{\kappa}_1$ (i.e. the proportion of topic 1:"Markets" in the corpus).

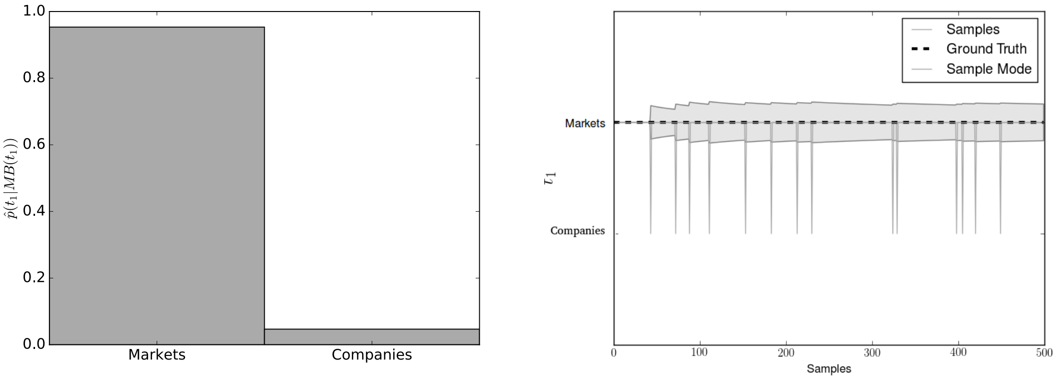
(d) Convergence of the mean of $\boldsymbol{\kappa}_2$ (i.e. the proportion of topic 2:"Companies" in the corpus).

Figure 3.5: Inference of the topic proportion $\boldsymbol{\kappa}$.

We observe that the vector of inferred topics also matches the ground truth. Finally, using the mean of $\boldsymbol{\pi}_j$ according to Equation 3.33 as the point estimate leads to

$$\overline{\mathbf{\Pi}^{(1)}} = \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix}, \quad \overline{\mathbf{\Pi}^{(2)}} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad \overline{\mathbf{\Pi}^{(3)}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The inferred confusion matrices indicate that the first annotator is an informative annotator (i.e. one that occasionally make mistakes in his judgment), the second annotator is a spammer (i.e. one that always selects the same topic with high confidence) and the third is a perfect annotator (i.e. one that never makes mistakes in his judgment).



(a) Empirical distribution of t_1 (i.e. the inferred topic of the first document).

(b) Convergence of the mode of t_1 (i.e. the inferred topic of the first document).

Figure 3.6: Inference of the topic t_1 for the first document.

3.4.6.2 Inference for Documents with Multiple Topics

We now assume that the ground truth of each document in the corpus is a mixture of the two topics. In particular, the ground truth for the topic proportion of the first document is given by the distribution

$$\mathbf{\Lambda}_1 = \left[\begin{array}{cc} 0.7 & 0.3 \end{array} \right].$$

That is, the topic 1:"Markets" is more than twice as relevant to the first document as the topic 2:"Companies". We consider the same set of $K = 3$ annotators providing the following judgments in the form of distributions for the first document

$$\Phi_1^{(1)} = \left[\begin{array}{cc} 0.5 & 0.5 \end{array} \right], \quad \Phi_1^{(2)} = \left[\begin{array}{cc} 1 & 0 \end{array} \right], \quad \Phi_1^{(3)} = \left[\begin{array}{cc} 0.7 & 0.3 \end{array} \right].$$

Given the above judgments $\Phi_1^{(k)}$, we are looking for the inferred posterior distribution $\hat{p}\left(t_1|\mathrm{MB}\left(t_1\right)\right)$ that best approximates the ground truth $\mathbf{\Lambda}_1$. To achieve this, we start by assigning the same uninformative uniform prior distribution for the topic proportion $\boldsymbol{\kappa}$ and each confusion matrix $\mathbf{\Pi}^{(k)}$. Next, we sample uniformly each annotators' judgment $\Phi_1^{(k)}$ a single time, giving the same implicit judgments $\mathbf{c}^{(k)}$ as before. Under the same initial conditions of the Markov chain as before, we run the Gibbs' sampling algorithm from Algorithm 3 using $\mathbf{c}^{(k)}$ as input. From Figure 3.6a, the inferred posterior distribution for t_1 is

$$\hat{p}\left(t_1|\mathrm{MB}\left(t_1\right)\right) = \left[\begin{array}{cc} 0.95 & 0.05 \end{array} \right].$$

That is, IBCC inferred that the first topic 1:"Markets" is 95% relevant to first document, while topic 2:"Companies" is only 5% relevant. It is clear from this example that IBCC does not accurately infer the ground truth $\mathbf{\Lambda}_1$. This is explained by the fact that a single sample from the topic proportion $\mathbf{\Lambda}_i$ of each document i (Line 4 in Algorithm 5) is not sufficient to recover a full distribution.

3.4.7 Empirical Evaluation

In this section, we empirically evaluate IBCC in aggregating judgments from documents with multiple topics. We assume that the topics have been generated from a topic model, and we further assume that the target topic of a document is the topic with the highest-probability in the topic-mixture distribution. We name this topic the *dominant topic*. We conduct an experiment to assess the accuracy of IBCC at recovering the dominant topic (Section 3.4.7.2) when each document has multiple topics in the presence of unreliable annotators. We simulate the ground truth and observations from an artificial set of annotators and assess the aggregation performance against the ground truth.

We first describe the experimental setting in Section 3.4.7.1. Next, we describe our accuracy metric in Section 3.4.7.2. Finally, we detail our results in Section 3.4.7.3.

3.4.7.1 Experimental Setting

We evaluate IBCC empirically on two synthetic datasets generated using Algorithm 1 and Algorithm 5. In particular, the first dataset contains only judgments from documents with a single target topic (the benchmark), while the second dataset contains only judgments from documents with multiple topics. Both datasets consist of judgments for $I = 50$ documents across $J = 10$ topics.

Each document in each dataset is judged by a number of annotators (ranging from 1 to 1,000), where each annotator is not allowed to judge the same document more than once. We simulate an informative annotator with bias toward the target topic by setting his error rate $\pi_{j,l}^{(k)}$ such that 75% of his judgments are correct across topics, that is

$$\pi_{j,l}^{(k)} = \begin{cases} 0.75 & \text{if } l = j, \\ \frac{0.25}{J-1} & \text{otherwise.} \end{cases}$$

For example, an informative annotator judging documents with three alternatives will have the following confusion matrix

$$\mathbf{\Pi}^{(k)} = \begin{bmatrix} 0.75 & 0.125 & 0.125 \\ 0.125 & 0.75 & 0.125 \\ 0.125 & 0.125 & 0.75 \end{bmatrix}.$$

Furthermore, a uniform spammer with strong bias towards a given topic is simulated by setting his error rate such that the same topic is always selected with probability one regardless of the document (i.e. a single column of his confusion matrix $\mathbf{\Pi}^{(k)}$ is set to one), that is

$$\pi_{j,l}^{(k)} = \begin{cases} 1 & \text{if } l = m^{(k)}, \\ 0 & \text{otherwise.} \end{cases}$$

The single topic $m^{(k)}$ that each uniform spammer spam on is drawn at random for each spammer, such that

$$m^{(k)} \sim \text{Uniform}\left(J\right).$$

For example, a uniform spammer judging documents with three alternatives when $m^{(k)} = 2$ will have the following confusion matrix

$$\mathbf{\Pi}^{(k)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Finally, we run IBCC four times on each dataset to ensure a confidence interval of less than a standard deviation.

3.4.7.2 Accuracy Metric

The ground truth associated with each document with a single topic is a single target topic, while the ground truth associated with each document with multiple topics is the topic with the highest probability (i.e. the dominant topic). To identify the dominant topic when documents have multiple topics, we use the mode of the document's topic proportion $\mathbf{\Lambda}_i$ (as defined in Section 3.4.5) such that,

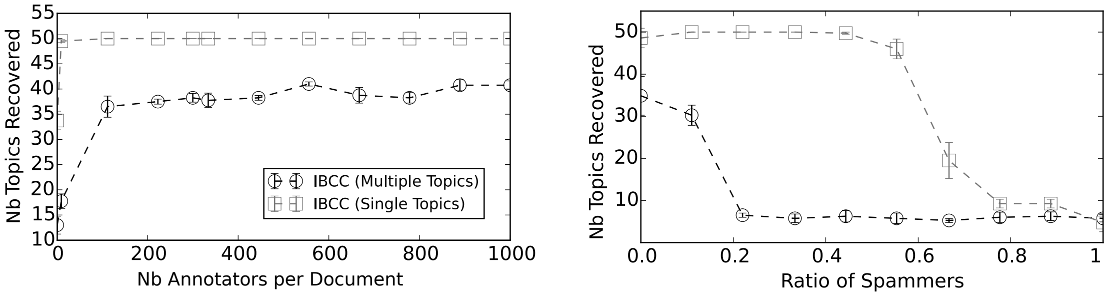
$$\text{mode}\left(\mathbf{\Lambda}_i\right) = t_i.$$

For example if a document has the following topic distribution $\mathbf{\Lambda}_i = \begin{bmatrix} 0.05 & 0.5 & 0.45 \end{bmatrix}$ the dominant topic will be the second topic as it has the highest probability of being selected by the annotators. Therefore for this document, the target topic $t_i = \text{mode}\left(\begin{bmatrix} 0.05 & 0.5 & 0.45 \end{bmatrix}\right)$ is equal to 2.

3.4.7.3 Results

We now present the outcome of the empirical evaluation of IBCC on the two datasets:

Aggregation Accuracy. Figure 3.7a shows the average number of target and dominant topics recovered when increasing the number of informative annotators per document from 1 to 1,000. From Figure 3.7a, we see that when documents have a single topic, IBCC recovers all the 50 targets topics on average when at least 10 different informative annotators provides a judgment. In contrast, when inferring the dominant topic from documents with multiple topic, IBCC fails to recover all the 50 documents. This happens even when 1,000 informative annotators provide judgment for a single document. Nevertheless, a number of dominant topics are still being recovered (about 37 on average out 50 for above 150 annotators per



(a) Average number of target and dominant topics recovered for various number of annotators providing a single judgment per document (with no spammers).

(b) Average number of target and dominant topics recovered for various ratios of spammers ranging from 0 to 100% (with 300 annotators providing a single judgment per document).

Figure 3.7: Average number of target and dominant topics recovered when: (a) varying the number of annotators per documents, and (b) varying the ratio of spammers.

document). This is explained by the fact that the dominant topic (i.e. the mode of $\mathbf{\Lambda}_i$) is more likely to be drawn from the topic distribution than the alternatives in each document with multiple topics (Algorithm 5 line 5).

Robustness to Spammers. Figure 3.7b shows the average number of target and dominant topics recovered when increasing the ratio of spammer from 0 to 100%, while setting the number of judgments per document to 300. We set the number of judgments to 300 because at this value the number of dominant topics recovered by IBCC on documents with multiple topics has converged on average at 37 out of 50 (Figure 3.7a). From Figure 3.7b, it takes 70% of spammers for IBCC to recover at least half of the target topics on average for documents with single topics. While for documents with multiple topics, 18% of spammers is enough to reduce the accuracy of IBCC by half. Therefore, when faced with documents with multiple topics, IBCC is 40% more sensitive to spammers when recovering half of the dominant topics.

3.5 Summary

This chapter introduced a number of methods for aggregating judgments from annotators of uneven quality. In particular, we presented non-weighted aggregation methods that do not take into account the annotators' accuracy. We then reviewed methods which take into account annotators' accuracy by means of a single weight. We reviewed a number of methods which weigh the annotators using confusion matrices, thus producing more accurate aggregations when faced with spammers than single weight methods. In Particular, we introduced the state-of-the-art model IBCC [Kim and Ghahramani, 2012] which make simplifying assumptions of independence to reduce the complexity of the model and the inference. We derived closed-formed solutions for the Gibbs' sampling algorithm for IBCC which enables it to make exact inference on the target topics, and the annotators' confusion matrix. Although Gibbs' sampling is relatively simple to derive and implement, when the number of documents and/or annotators is large (as we will see in Chapter 4), such an approach is no longer appropriate due to its low rate of convergence. Consequently, we derived the closed formed solution for the optimal variational equations of IBCC in Section 3.4.4, which provide fast but approximate inference. Since we aim at aggregating judgments for documents with multiple topics (Aim 1), we extended the generative process for documents with a single topic assumed by IBCC, to documents with multiple topics in Section 3.4.5. Next, we outlined a concrete scenario where inference is performed on a corpus of documents with single and multiple topics using the Gibbs' sampling algorithm. We showed that when IBCC is faced with documents with multiple topics (which are defined as a mixture probability distribution over topics), it is unable to accuratly infer the aggregation and the annotators' confusion matrix. Finally, we evaluated IBCC experimentally in aggregating judgments for documents with multiple topics. Results on synthetic data showed that when faced with judgments from documents with multiple topics, IBCC fails to take this fact into consideration, leading to inaccurate aggregation. While the greater modelling power of confusion matrices can compensate for the annotators' bias when documents have a single topic, they do not easily generalise when multiple target topics can be assigned to a single document. This limitation provides our next challenge for extending IBCC to handle documents with multiple topics as it will be discussed in the next chapter.

Chapter 4

Evaluating Topic Models by Aggregating Human Judgments

Although IBCC performs well in terms of accuracy of classification when documents have a single topic, we showed empirically in Chapter 3 that it is not well suited in the case when documents have multiple topics. In this chapter we introduce and evaluate our contribution fully addressing Aim 1. In particular, we build upon IBCC to directly address the problem of assessing the relevance of documents against a set of topics discovered by topic models. We first detail our model in Section 4.1 which extends IBCC and aggregates probability distributions over multiple topics. Since our approach is computationally more expensive than IBCC, we use variational inference in Section 4.2. We then give a step-by-step example in Section 4.3 to illustrate how evaluates topical relevance from the annotators. Finally, in Section 4.4 we evaluate the performance of our model empirically against state-of-the-art approaches (including opinion pools and confusion-matrix-based approaches) in an aggregation scenario with multiple categories. We end the chapter with an overall summary.

4.1 Model Description

Our model – referred to hereafter as multi-category independent Bayesian classifier combination (MBCC) – is a generalisation of IBCC that deals with settings where documents have multiple categories. In more detail, we introduce a categorical distribution (i.e. the topic proportion) with parameter $\mathbf{\Lambda}_i$ over the J topics for each document i (Equation 4.1) instead of the single categorical distribution $\boldsymbol{\kappa}$ common to all documents found in IBCC (Equation 3.13). Moreover, each annotator k submits a distribution $\Phi_i^{(k)}$ over the J topics for each document i instead of the single topic $c_i^{(k)}$ required by IBCC. The confusion matrix $\mathbf{\Pi}^{(k)}$ of each annotator k is kept unchanged from IBCC. Therefore in this new setting, to assess the accuracy of each annotator through their confusion matrices

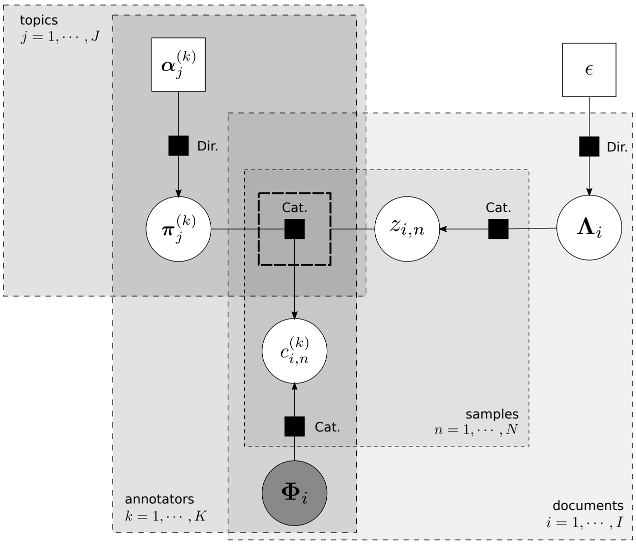


Figure 4.1: Factor graph of MBCC.

$\boldsymbol{\Pi}^{(k)}$, we need to first independently sample the aggregated distribution $\boldsymbol{\Lambda}_i$ to obtain a set of topics $z_{i,n}$ for each document i such that

$$z_{i,n} \sim \operatorname{Cat}\left(\boldsymbol{\Lambda}_i\right), \tag{4.1}$$

for all samples $n \in \{1, \cdots, N\}$. The vector $\mathbf{z}_i$ of dimension N can be seen as independent target topics t_i of document i (as in IBCC) drawn from the topic proportions in Equation 4.1. We subsequently match these samples against samples from each of the annotators' distribution $\Phi_i^{(k)}$ to obtain

$$c_{i,n}^{(k)} \sim \operatorname{Cat}\left(\boldsymbol{\pi}_{z_{i,n}}^{(k)}\right) \tag{4.2}$$

for each document i. This is equivalent to running IBCC N times with different values of the annotators' judgment $c_i^{(k)}$ drawn from their distribution $\Phi_i^{(k)}$. In practice, one can perform this sampling until the accuracy no longer increases. Alternatively, a theoretical upper bound for an arbitrary level of error can be calculated using Lemma 3 in [Devroye, 1983]. Thus, the joint distribution is

$$\begin{aligned} p\left(\mathbf{z}, \mathbf{c}, \boldsymbol{\Lambda}, \boldsymbol{\Pi}\right) \quad = \quad & \prod_{i=1}^{I} \operatorname{Cat}\left(\boldsymbol{\Lambda}_i\right) \operatorname{Dir}\left(\epsilon_i\right) \times \\ & \prod_{k=1}^{K} \prod_{n=1}^{N} \operatorname{Cat}\left(\boldsymbol{\pi}_{z_{i,n}}^{(k)}\right) \prod_{j=1}^{J} \operatorname{Dir}\left(\boldsymbol{\alpha}_j^{(k)}\right), \end{aligned} \tag{4.3}$$

where the hyperparameter ν in IBCC has been renamed for clarity to ϵ_i (i.e. the pseudo-count of topics for each document i), such that

$$\boldsymbol{\Lambda}_i \sim \operatorname{Dir}\left(\epsilon_i\right). \tag{4.4}$$

Algorithm 6 Generative process expressing hypotheses about the way in which the annotators' judgment for each documents have been generated.

```
1: Input: the confusion matrices $\mathbf{\Pi}$ and the category proportions $\mathbf{\Lambda}$
2:
3: for each document $i \in \{1, \cdots, I\}$ do
4:     for each sample $n \in \{1, \cdots, N\}$ do
5:         Sample $z_{i,n} \sim \mathrm{Cat}(\mathbf{\Lambda}_i)$
6:         for each annotator $k \in \{1, \cdots, K\}$ do
7:             Sample $c_{i,n}^{(k)} \sim \mathrm{Cat}\left(\boldsymbol{\pi}_{z_{i,n}}^{(k)}\right)$
8:         end for
9:     end for
10:
11:     for each annotator $k \in \{1, \cdots, K\}$ do
12:         for each category $j \in \{1, \cdots, J\}$ do
13:             $\Phi_{i,j}^{(k)} = \beta_{i,j}^{(k)} \sum_{n=1}^{N} \delta\left(c_{i,n}^{(k)} - j\right)$
14:             where $\beta_{i,j}^{(k)}$ is a normalising constant.
15:         end for
16:     end for
17: end for
18:
19: return $\mathbf{\Phi}$
```

The generative process, that is, the random process by which MBCC assumes the annotators' judgment $\Phi_i^{(k)}$ arose, is summarised in Figure 4.1 and Algorithm 6. In particular, we start with the confusion matrices $\mathbf{\Pi}$, and the topic proportions $\mathbf{\Lambda}$ sampled from Equations 3.16 and 4.4 respectively (Line 1). We then sample N topics $z_{i,n}$ from each topic proportion $\mathbf{\Lambda}_i$ from Equation 4.1 (Line 5). Given each topic $z_{i,n}$, we sample judgments $c_{i,n}^{(k)}$ from the annotators' $z_{i,n}$-th row of their confusion matrix $\mathbf{\Pi}^{(k)}$ (Line 7). Finally, we find the most likely categorical distributions $\Phi_i^{(k)}$ which generated the samples $\mathbf{c}_i^{(k)}$ for all documents i and annotators k (Line 13).

The key inferential problem that needs to be solved in order to use MBCC is that of computing the posterior distribution of the latent variables $\mathbf{z}$, and the parameters $\mathbf{\Lambda}$ and $\mathbf{\Pi}$ given the data $\mathbf{c}$, that is $p\left(\mathbf{z}, \mathbf{\Lambda}, \mathbf{\Pi} | \mathbf{c}\right)$.

4.2 Inference using Variational Methods

Building on the variational equations from IBCC in Section 3.4.4.2, we replace the variables specific to IBCC with their counterpart in MBCC. The hyperparameter for the topic proportion of the corpus $\boldsymbol{\nu}$ in IBCC is replaced with the hyperparameter of the topic proportion per document $\boldsymbol{\epsilon}$ in MBCC

$$\boldsymbol{\nu} \rightarrow \boldsymbol{\epsilon}.$$

Algorithm 7 Variational inference for MBCC.

```
Inputs: nb iterations $M > 0$, $\boldsymbol{\epsilon}$, $\mathbf{A}$, $\mathbf{c}$

Initialise $E_{\mathbf{\Pi}}\left[\ln\pi_{j,l}^{(k)}\right]^{[0]}$ and $E_{\mathbf{\Lambda}}\left[\ln\Lambda_{i,j}\right]^{[0]}$.

for each iteration $m$ in $\{1, \cdots, M\}$ do
    for each sample $n$ in $\{1, \cdots, N\}$ do
        for each document $i$ in $\{1, \cdots, I\}$ do
            Compute $E_{\mathbf{z}}\left[z_{i,n} = j\right]^{[m]}$
        end for
    end for

    for each document $i$ in $\{1, \cdots, I\}$ do
        for each topic $j$ in $\{1, \cdots, J\}$ do
            Compute $E_{\mathbf{\Lambda}}\left[\ln\Lambda_{i,j}\right]^{[m]}$
        end for
    end for

    for each annotator $k$ in $\{1, \cdots, K\}$ do
        for each topic $j$ in $\{1, \cdots, J\}$ do
            for each topic $l$ in $\{1, \cdots, L\}$ do
                Compute $E_{\mathbf{\Pi}}\left[\ln\pi_{j,l}^{(k)}\right]^{[m]}$
            end for
        end for
    end for
end for

return $q^*(\mathbf{z})$, $q^*(\mathbf{\Lambda})$, and $q^*(\mathbf{\Pi})$
```

Since each document has now its own topic proportion, we do the following substitution

$$\boldsymbol{\kappa} \rightarrow \prod_{i=1}^{I} \mathbf{\Lambda}_i.$$

Each target topic is now a series of samples from the topic proportion per document

$$t_i \rightarrow \prod_{n=1}^{N} z_{i,n}.$$

Each annotators' judgment becomes a set of samples from their distribution

$$c_i^{(k)} \rightarrow \prod_{n=1}^{N} c_{i,n}^{(k)}.$$

Finally, the hyperparameters $\mathbf{A}$ and the confusion matrices $\mathbf{\Pi}$ are left unchanged. The resulting modified algorithm for MBCC is given in Algorithm 7.

4.3 Worked Example

Building on the worked example in Section 3.4.6, we illustrate the use of MBCC at aggregating the annotators' judgments regarding a corpus of documents with multiple topics. The exemplar corpus is the same set of news articles from Section 3.4.6.2. As before, we will particularly pay attention to the first document which will serve as a representative baseline. This time however, we sample the distribution $\Phi_1^{(k)}$ for the first document of each annotator $N = 10$ times instead of a single time in IBCC (Line 4 in Algorithm 6, instead of Line 6 in Algorithm 5), giving

$$\mathbf{c}_1^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 2 & 1 & 2 & 2 & 2 & 1 & 2 \end{bmatrix}$$

for the first annotator,

$$\mathbf{c}_1^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

for the second annotator, and

$$\mathbf{c}_1^{(3)} = \begin{bmatrix} 1 & 1 & 1 & 2 & 1 & 2 & 1 & 1 & 1 & 2 \end{bmatrix}$$

for the third annotator. Each vector of samples $\mathbf{c}_1^{(k)}$ is used internally by MBCC as repeated implicit judgments from a single annotator k for the first document in a similar way as IBCC. Now, to infer the posterior distribution of each latent variable $\mathbf{\Lambda}$, $\mathbf{\Pi}$ and $\mathbf{z}$ given $\mathbf{c}$, we use variational inference (Section 3.4.4) as N can be arbitrarily large. To do this, we step through Algorithm 7.

Line 1 We first initialise the hyperparameters $\mathbf{A}$ and $\boldsymbol{\epsilon} = \boldsymbol{\nu}$ as before, and arbitrarily set the number of iterations of the variational inference to $M = 50$.

Line 3 We initialise the expectation of the logarithm of the model's parameters using arbitrary values. For simplicity, we re-use the initial values from the worked example of IBCC, giving

$$E_{\mathbf{\Pi}}\left[\ln \pi_{j,l}^{(k)}\right]^{[0]} = \pi_{j,l}^{(k)[0]}$$

for the confusion matrices, and

$$E_{\mathbf{\Lambda}}\left[\ln \Lambda_{i,j}\right]^{[0]} = \boldsymbol{\kappa}_j^{[0]} \text{ for all } i$$

for the topic proportion per documents.

Line 7 To compute the expectation $E_{\mathbf{z}}\left[z_{1,1} = 1\right]^{[1]}$, that is, the expectation that the first sample n of the topic proportion $\mathbf{\Lambda}_i$ for the first document i at the first iteration m is the topic 1:"Markets", we first need to compute the intermediary term given by Equation

3.42

$$\begin{aligned}
\rho_{1,1}^{[1]} &= \exp\left(E_{\mathbf{\Lambda}}\left[\ln\Lambda_{1,1}\right]^{[0]} + \sum_{k=1}^{3} E_{\mathbf{\Pi}}\left[\ln\pi_{1,c_{1,1}^{(k)}}^{(k)}\right]^{[0]}\right) \\
&= \exp\left(0.23 + (0.47 + 0.13 + 0.51)\right) = 3.8.
\end{aligned}$$

And similarly for the topic 2:“Companies”,

$$\begin{aligned}
\rho_{1,2}^{[1]} &= \exp\left(E_{\mathbf{\Lambda}}\left[\ln\Lambda_{1,2}\right]^{[0]} + \sum_{k=1}^{3} E_{\mathbf{\Pi}}\left[\ln\pi_{2,c_{1,1}^{(k)}}^{(k)}\right]^{[0]}\right) \\
&= \exp\left(0.77 + (0.72 + 0.77 + 0.87)\right) = 22.87.
\end{aligned}$$

Combining the two terms using Equation 3.43 gives the expectations

$$E_{\mathbf{z}}\left[z_{1,1} = 1\right]^{[1]} = \frac{\rho_{1,1}^{[1]}}{\rho_{1,1}^{[1]} + \rho_{1,2}^{[1]}} = \frac{3.8}{3.8 + 22.87} = 0.14,$$

and

$$E_{\mathbf{z}}\left[z_{1,1} = 2\right]^{[1]} = \frac{\rho_{1,2}^{[1]}}{\rho_{1,1}^{[1]} + \rho_{1,2}^{[1]}} = \frac{22.87}{3.8 + 22.87} = 0.86.$$

From Equation 3.44, the optimal approximate probability distribution of the first sample of the topic proportion for the first document at the first iteration is given by

$$q^*\left(z_{1,1} = j\right)^{[1]} = \begin{bmatrix} 0.14 & 0.86 \end{bmatrix}.$$

We proceed similarly across all the samples n and documents i.

Line 15 Next, to compute $E_{\mathbf{\Lambda}}\left[\ln\Lambda_{1,1}\right]^{[1]}$ and $E_{\mathbf{\Lambda}}\left[\ln\Lambda_{1,2}\right]^{[1]}$, that is, the expectation of the log-probability of the topic 1:“Markets” and 2:“Companies” in the first document at the first iteration, we first need to compute $\mathbf{N}_1^{[1]}$, the expectation of the number of occurrences of each topic j in $\mathbf{t}$ at the first iteration for the first sample n. Using Equation 3.45 we get

$$\mathbf{N}_1^{[1]} = \begin{bmatrix} 2 & 1 \end{bmatrix}.$$

Then, following Equation 3.49, we get

$$\begin{aligned}
E_{\mathbf{\Lambda}}\left[\ln\mathbf{\Lambda}_{1,1}\right]^{[1]} &= \psi\left(\epsilon_{1,1}^{[0]} + N_1^{[1]}\right) - \psi\left(\sum_{\iota=1}^{2}\epsilon_{1,\iota}^{[0]} + N_1^{[1]}\right) \\
&= \psi\left(1+2\right) - \psi\left((1+2) + (1+1)\right) \\
&= 0.92 + 1.5 = 2.42.
\end{aligned}$$

Proceeding similarly for the topic 2:"Companies" of the first document, we get

$$\begin{aligned} E_{\mathbf{\Lambda}}\left[\ln\mathbf{\Lambda}_{1,2}\right]^{[1]} &= \psi\left(\epsilon_{1,2}^{[0]}+N_2^{[1]}\right)-\psi\left(\sum_{\iota=1}^{2}\epsilon_{1,\iota}^{[0]}+N_1^{[1]}\right) \\ &= \psi\left(1+1\right)-\psi\left(\left(1+2\right)+\left(1+1\right)\right) \\ &= 0.42+1.5=1.92 \end{aligned}$$

Re-writing the two results as a vector gives,

$$E_{\mathbf{\Lambda}}\left[\ln\mathbf{\Lambda}_1\right]^{[1]}=\left[\begin{array}{cc} 2.42 & 1.92 \end{array}\right].$$

We repeat this step for all topics j and documents i.

Line 22 Next, we need to compute $E_{\mathbf{\Pi}}\left[\ln\pi_{1,2}^{(1)}\right]^{[1]}$ and $E_{\mathbf{\Pi}}\left[\ln\pi_{1,1}^{(1)}\right]^{[1]}$. The expectation $E_{\mathbf{\Pi}}\left[\ln\pi_{1,2}^{(1)}\right]^{[1]}$ is the expectation of the log-probability of the error rate of the first annotator when the first sample n of each document i from his judgment $\mathbf{c}_1^{(k)}$ is 2:"Companies" while the sample topic $z_{1,1}$ is 1:"Markets". To do this, we first need to compute $\mathbf{M}_1^{(1)[1]}$ using Equation 3.46

$$\mathbf{M}_1^{(1)[1]}=\left[\begin{array}{cc} 6 & 4 \end{array}\right].$$

Using Equation 3.51, we get

$$\begin{aligned} E_{\mathbf{\Pi}}\left[\ln\pi_{1,1}^{(1)}\right]^{[1]} &= \psi\left(\alpha_{1,1}^{(1)[0]}+M_{1,1}^{(1)[1]}\right)-\psi\left(\sum_{l=1}^{L}\alpha_{1,l}^{(1)[0]}+M_{1,l}^{(1)[1]}\right) \\ &= \psi\left(1+6\right)-\psi\left(\left(1+6\right)+\left(1+4\right)\right) \\ &= 1.87-2.44=-0.57, \end{aligned}$$

and

$$\begin{aligned} E_{\mathbf{\Pi}}\left[\ln\pi_{1,2}^{(1)}\right]^{[1]} &= \psi\left(\alpha_{1,2}^{(1)[0]}+M_{1,2}^{(1)[1]}\right)-\psi\left(\sum_{l=1}^{L}\alpha_{1,l}^{(1)[0]}+M_{1,l}^{(1)[1]}\right) \\ &= \psi\left(1+4\right)-\psi\left(\left(1+6\right)+\left(1+4\right)\right) \\ &= 1.5-2.44=-0.94. \end{aligned}$$

Re-writing the two results as a vector gives,

$$E_{\mathbf{\Pi}}\left[\ln\boldsymbol{\pi}_1^{(1)}\right]^{[1]}=\left[\begin{array}{cc} -0.67 & -0.94 \end{array}\right].$$

We repeat this step for all annotators k, all topics j and sample $c_{i,n}^{k)}$ of value l.

Line 25 After each iteration m starting at Line 5, each expectation term can be used to check convergence. Alternatively, the lower bound $\mathcal{L}\left(q\right)$ can also be calculated according to Equation 3.36 to check for convergence. Once the value of the lower bound $\mathcal{L}\left(q\right)$ has

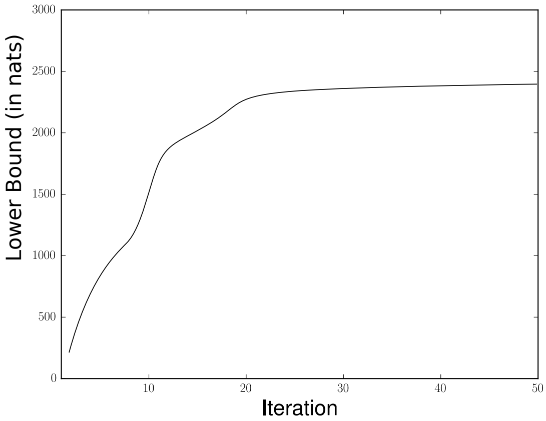


Figure 4.2: Illustrative example of the evolution of the evidence lower bound $\mathcal{L}(q)$ of MBCC during inference. Since the logarithms are taken to base e, the lower bound is measured in natural unit of information (nats).

stopped increasing, the algorithm has converged to the optimal approximate solution. Figure 4.2 shows the convergence for $M = 50$ iterations.

Line 27 After convergence, the algorithm returns the approximate posterior distribution $q^*(.)$ for each variable, from which we can take the expected value (Equation 3.52, 3.53, and 3.54). For the topic proportion of the first document, using Equation 3.53 we get

$$E_{\boldsymbol{\Lambda}}\left[\boldsymbol{\Lambda}_1^{[M]}\right] = \left[\begin{array}{cc} 0.68 & 0.32 \end{array}\right] \tag{4.5}$$

Since each topic proportion $\boldsymbol{\Lambda}_i$ is unique to each document, we are able to recover the topic proportion individually for each document. In contrast, in IBCC, the topic proportion $\boldsymbol{\kappa}$ is shared across the documents in the corpus and does not allow us to capture the topic distribution of each document. Furthermore, the expectation of the approximate posterior distribution for the confusion matrix of each annotator (Equation 3.54) is given by

$$E_{\boldsymbol{\Pi}}\left[\boldsymbol{\Pi}^{(1)[M]}\right] = \left[\begin{array}{cc} 0.72 & 0.28 \\ 0.26 & 0.74 \end{array}\right], \quad E_{\boldsymbol{\Pi}}\left[\boldsymbol{\Pi}^{(2)[M]}\right] = \left[\begin{array}{cc} 0.99 & 0.01 \\ 0.98 & 0.02 \end{array}\right],$$

$$E_{\boldsymbol{\Pi}}\left[\boldsymbol{\Pi}^{(3)[M]}\right] = \left[\begin{array}{cc} 0.98 & 0.02 \\ 0.01 & 0.99 \end{array}\right].$$

The inferred confusion matrices indicate that the first annotator is an informative annotator (i.e. one that occasionally make mistakes in his judgment $\boldsymbol{\Phi}^{(1)}$), the second annotator is a spammer (i.e. one that always selects the same topic with high confidence) and the third is a perfect annotator (i.e. one that never makes mistakes in his judgment $\boldsymbol{\Phi}^{(3)}$). It is important to note that these inferred values are approximations of the ground truth since, unlike Gibbs' sampling, variational inference is not an exact inference algorithm. In particular, we can use the KL-divergence to compare the loss of information from the inferred topic proportion $E_{\mathbf{\Lambda}}\left[\mathbf{\Lambda}_1^{[M]}\right]$ for the first document in Equation 4.5 to the ground truth, that is

$$\begin{aligned} D_{KL}\left(\mathbf{\Lambda}_1 || E_{\mathbf{\Lambda}}\left[\mathbf{\Lambda}_1^{[M]}\right]\right) &= \sum_{j=1}^{J} \mathbf{\Lambda}_{1,j} \ln \frac{\mathbf{\Lambda}_{1,j}}{E_{\mathbf{\Lambda}}\left[\mathbf{\Lambda}_{1,j}^{[M]}\right]} \\ &= 0.7 \ln \frac{0.7}{0.68} + 0.3 \ln \frac{0.3}{0.32} \\ &= 9.3 \times 10^{-4}. \end{aligned}$$

Since the divergence is negligible, the algorithm has inferred the ground truth within acceptable margin of accuracy.

4.4 Empirical Evaluation

To evaluate the efficacy of our model, we use an independently gathered dataset, and introduce two new datasets; all of which include ground truth from expert annotators. We then compare performance against four state-of-the-art benchmarks. The experiments are run in an unsupervised setting, where the ground truth is never exposed to the algorithms, and is only used to measure their accuracy.

4.4.1 Datasets

Our experiments use a total of three datasets.

SemEval. This dataset contains judgments of sentiments within one hundred news headlines sampled from the SemEval2007 test set [Strapparava and Mihalcea, 2007, Snow et al., 2008]. Each annotator was presented with a list of headlines, and was asked to give numeric judgments between zero and a hundred for each of six sentiments: *joy, sadness, disgust, anger*, *surprise,* and *fear*. Figure 4.3 illustrates this dataset[1]. Ten judgments were collected for each headline for a total of 1,000 judgments. These judgments were obtained from 38 annotators whom provided a minimum of 20 judgments each, and 26 on average. We truncate the total number of judgments per annotator to 20 to avoid discrepancies in accuracy of each inferred confusion matrix. We normalise the values submitted by each annotator into valid probability distributions by ensuring that the total area is equal to one at any given time.

IAPR-TC12. We crowdsourced a set of 16 images sampled from the IAPR-TC12 dataset [Escalante et al., 2010, Augustin and Venanzi, 2017]. This is a collection of 20,000 images of urban and rural scenes manually segmented per region. Each pixel

[1] A sample of 128 jugdments from the test set can be found in Appendix B

```
Task 55: Making peace from victory over poverty.
Joy  [70]   Sadness  [0] Disgust [35]
Anger [0]   Surprise [0] Fear     [0]

Task 591: Iran says it will strike US interests if Attacked.
Joy     [0]   Sadness  [100] Disgust  [100]
Anger [100]   Surprise   [0] Fear      [40]

Task 592: Palestinian factions to resume talks.
Joy   [0]   Sadness  [0] Disgust  [0]
Anger [0]   Surprise [0] Fear     [0]
```

Figure 4.3: Example of annotated news headline taken from the SemEval2007 dataset.

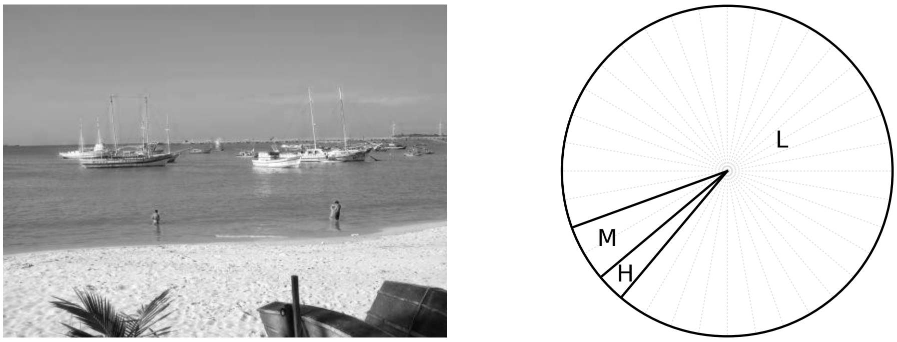


Image 61: Fill the pie chart to reflect the proportion
of the image to each of the following categories:
[H]uman(s), [A]nimal(s), [F]ood,
[L]andscape-Nature, [M]an-made, [O]ther.

Figure 4.4: Example of a judgment in a rural scene from the IAPR-TC12 dataset performed with a pie chart.

belongs to one of six region. We gathered a total of 21 judgments per image from 21 annotators. Workers were asked to estimate the proportion of each region in the image. The ground truth proportion for each category is calculated by dividing the number of pixels in the region by the total number of pixels in the image. The annotators reported their judgments with a pie chart, enabling quick and accurate judgments of proportion [Hollands and Spence, 1992]. Figure 4.4 illustrates this dataset.

Colours. We crowdsourced a set of 460 judgments of the proportion of colours in the flags of 20 countries [Augustin and Venanzi, 2017]. We asked 23 participants to judge, from memory, the proportion of 10 colours in each country's flag.

The alternatives in each dataset are complete, that is, using all the categories can always fully describe the instance. Furthermore, although these three datasets may already contain spammers, we augment the datasets with additional synthetic spammers to explore the loss of accuracy as they increase in number. While the obvious characteristic of

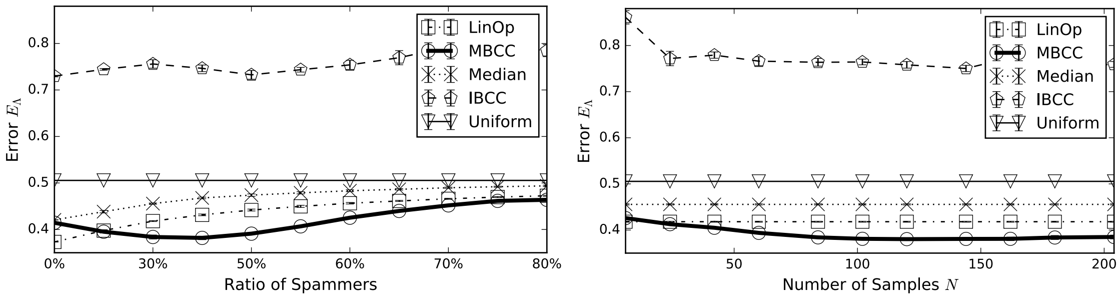


Figure 4.5: Average error on the aggregated distributions $\mathbf{\Lambda}$ on the SemEval dataset when increasing: (left) the ratio of spammers at $N = 180$ samples, (right) the number of samples at a ratio of spammers of 50%.

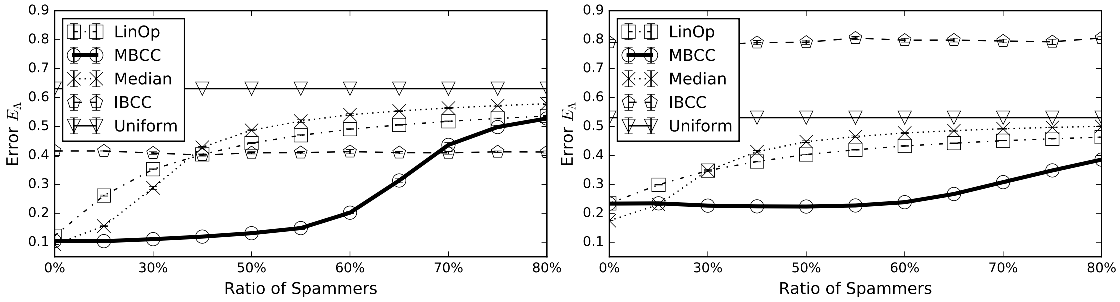


Figure 4.6: Average error on the aggregated distributions when increasing the ratio of spammers on: (left) the IAPR-TC12 dataset at $N = 180$ samples, (right) the Colours dataset at $N = 330$ samples.

repeating judgment patterns can be used to manually filter out uniform spammers, random spammers are the most challenging to detect (Section 3.3). For this reason, we use a random spamming strategy where a spammer always provides a random categorical distribution for each document. The distribution of each spammer k for each document i shares a prior Dirichlet distribution such that

$$\Phi_i^{(k)} \sim \text{Dir}\left(\mathbf{1}\right), \tag{4.6}$$

where the pseudo-count of $\mathbf{1}$ ensures that all possible distributions $\Phi_i^{(k)}$ are equally likely.

4.4.2 Experimental Setting

We set the parameter of the prior probability of each confusion matrix for all annotators and spammers to $\mathbf{A}^{(k)} = 100 \times \mathbf{I} + \mathbf{1}^T\mathbf{1}$. This means that annotators are initially assumed to be reasonably accurate before seeing any data. Using a different assumption leads to distinct aggregation accuracy profiles that will be discussed in more detail in Section 4.4.5. Furthermore, to ensure fair comparison between the benchmarks, we do not adjust each parameters $\boldsymbol{\epsilon}$ of the prior distributions over categories to reflect the balance of each dataset. Finally, we run all models a hundred times each to achieve statistically significant results at the 99% confidence level.

4.4.3 Benchmarks

We compare the performance of our model to four state of the art benchmark methods.

Uniform assigns a uniform distribution for each aggregated distribution (i.e. $p_j = \frac{1}{J}$ for all $j \in \{1, \cdots, J\}$), making it independent of the dataset. These particular values of p_j maximise the entropy function of categorical distributions. That is, if one were to guess the aggregated distribution far from the ground truth on average (given some distance metric), it can be expected to have its error above the uniform model.

LinOp averages the distributions provided by the annotators. Unlike IBCC and MBCC, it does not sample the annotators' distributions to produce the discrete observations, but directly takes the distributions as input. As there is no training set for assigning informative weights to the annotators, we assign equal weights $\omega^{(k)} = \frac{1}{K}$ to each annotator.

Median estimates the aggregated distribution by arranging the judgments for each document in ascending order and then takes the middle judgment. Each judgment is considered with equal weight for the same reason as for LinOp. The median is a robust method against extreme judgments since it will not give an arbitrarily large or small result if no more than half of the judgments for a document are incorrect.

IBCC combines discrete judgments from multiple annotators and models the ability of each individual annotator using confusion matrices as defined in Equation 3.12. Although IBCC takes a single category as ground truth, the output is a posterior distribution over the categories which can be compared to the ground truth category proportion.

4.4.4 Performance Metrics

To assess the accuracy of the inference, we use the Euclidean distance. Specifically, we define the average error of the aggregation on the entire dataset by

$$E_{\mathbf{\Lambda}} = \frac{1}{I} \sum_{i=1}^{I} \mathrm{d}\left(\mathbf{\Lambda}_i^*, \mathbf{\Lambda}_i\right) \tag{4.7}$$

where $\mathrm{d}\left(.\right)$ is the Euclidean distance, I the total number of documents, $\mathbf{\Lambda}_i^*$ the ground truth distribution for document i, and $\mathbf{\Lambda}_i$ the aggregated distribution provided by the model for document i. Furthermore, we define the deviation of the confusion matrix

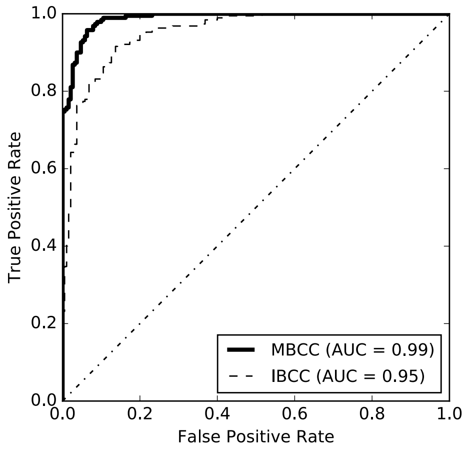


Figure 4.7: ROC curves on the SemEval dataset. The ratio of spammers is set to 50%.

$\mathbf{\Pi}^{(k)}$ of annotator k to the identity matrix $\mathbf{I}$ by

$$E_{\mathbf{\Pi}^{(k)}} = \sum_{j=1}^{J} \mathrm{d}\left(\mathbf{I}_j, \boldsymbol{\pi}_j^{(k)}\right). \tag{4.8}$$

Other distance metrics could be used, provided that they give finite results when the distributions are not absolutely continuous.

4.4.5 Results

We now present the results of our empirical evaluation regarding a number of key aspects: (i) accuracy of aggregation and robustness to spammers, (ii) convergence and running time, (iii) classification of spammers. We first consider in detail the results on the SemEval dataset, and discuss the results on the other datasets.

Figure 4.5 (left) shows the average error as the ratio of spammers increases on the SemEval dataset. To illustrate, a ratio of spammers of 50% means half the annotators in the dataset are spammers. As can be seen, MBCC achieves a comparable aggregation accuracy when 50% of the annotators are spammers, as LinOp does when no spammers are added. However, the added complexity of detecting spammers in MBCC comes at a cost when the level of spammers is low. Indeed, the highest precision is initially produced by LinOp, but as the ratio of spammers increases a crossover point is reached after which MBCC is the most precise. The value of this crossover point can be adjusted by varying the pseudo-counts of prior observations $\mathbf{A}_{ii}^{(k)}$ on the diagonal of the annotators' confusion matrix. For high values, MBCC assumes all annotators are truthful. This is the assumption underpinning LinOp, where all annotators have an implicit identity matrix as their confusion matrix. In fact, for high pseudo-counts, the performance of

MBCC matches perfectly with LinOp regardless of the number of spammers. On the other hand, with lower values of pseudo-counts, MBCC allows greater flexibility to learn the spammers, at the cost of having a greater error when there are fewer spammers in the dataset. Therefore, the added degrees of freedom lead to a tradeoff in accuracy at different ratio of spammers. Furthermore, we set the number of samples N empirically. Figure 4.5 (right) shows the average error of the aggregated distribution when increasing the number of samples at a ratio of spammers of 50%. As the number of samples increases, we observe convergence of the error at 100 samples to values of 0.75 and 0.37 for IBCC and MBCC respectively. Inevitably, the running time also increases as the number of samples increases. In particular, the running time of LinOp and Median is typically 3ms, while that of IBCC and MBCC ranges from 12s and 13s, to 6min and 28min respectively, across the range of samples shown in Figure 4.5 (right).

We now evaluate the accuracy of the confusion matrix-based models at classifying annotators from spammers on the SemEval dataset. To do this, we first collect the inferred confusion matrix $\mathbf{\Pi}^{(k)}$ for each annotator. We then compute the deviation $E_{\mathbf{\Pi}^{(k)}}$ of each confusion matrix to the identity matrix (Equation 4.8). The identity matrix represents the confusion matrix of a perfect annotator, that is, one that always gives judgments in accordance with the consensus. We then set a threshold on the error, above which a annotator is classified as a spammer. The receiver operating characteristic (ROC) curves in Figure 4.7 capture the effect of varying the threshold of the error. As can be seen, MBCC has an area under the curve (AUC) of 0.99, showing a 5 times improvement compared to IBCC in terms of the expected number of misclassified spammers. The AUC eventually decreases for both models at higher ratios.

We now discuss the results on the two other datasets. On the IAPR-TC12 dataset (Figure 4.6 (left)), LinOp, Median and MBCC achieve equal accuracies when no spammers are added. However, since each image in this dataset is dominated by a single category, that is, at least one category has more than 50% coverage in each image, IBCC performs reasonably well even with a high ratio of spammers. This is because IBCC always assigns a probability of one to the most likely category, which emphasises its assumption that documents have exactly one category. In fact, as the documents' proportion regresses to Kronecker deltas, the error from IBCC decreases. On the Colours dataset (Figure 4.6 (right)), the accuracy of MBCC remains a lower bound of LinOp throughout the range of ratios of spammers. However, Median initially achieves the lowest error with a crossover point with MBCC at 15% spammers. This is because the judgments provided by the annotators include sufficient outliers, making the Median a good measure of central tendency. Finally, convergence of the error on the aggregation is reached at 100 samples for the IAPR-TC12 dataset and 150 for the Colours dataset. Since the Colour dataset has 4 more categories than the IAPR-TC12 datasets, it requires more samples to achieve convergence.

4.5 Summary

In an effort to evaluate the relevance of the topics identified by topic models to human (Aim 1), we introduce a novel method of aggregating judgments of topic proportion from non-expert collected via crowdsourcing. Our proposed method, namely MBCC, addresses the shortcomings of IBCC (Chapter 3) and allows the evaluation of documents with a mixture of multiple topics (Contribution 2). In particular, our approach introduces a number of key innovations. First, we regard the problem of judging proportions as one of probabilistic modelling where the responses elicited from the participants are assumed to be categorical distributions. This has the benefit of taking advantage of the flexibility of the probabilistic framework in terms of intuitive causal modelling, model fitting, model selection and complexity reduction. Second, by relying on Bayesian methods, we conveniently avoid overfitting (Aim 4) by taking into account the uncertainty when making inference about the aggregation, and the participants' ability as expressed by their confusion matrices. Third, we account for the presence of malicious participants, who provide judgments randomly (i.e. spammers), by sampling and weighting their judgments in the aggregation via their confusion matrices. Our empirical evaluation based on three real-world datasets shows that our approach outperforms existing methods by up to 28% in terms of accuracy. We show a comparable level of accuracy when 60% of the annotators are spammers, as other approaches do when there are no spammers. Finally, we improve the expected number of misclassified spammers by up to five times that achieved by existing methods.

Chapter 5

Topic Learning from Sentence Embeddings

In the previous chapter, we introduced a new crowdsourcing model to support large-scale human assessment of the relevance of topics discovered by topic models. In this chapter, we address the distinct but related challenge of finding a novel solution to address topic interpretability (Aim 2). Our approach combines a sentence embedding model and a hard clustering model to identify topics consisting of sentences. In particular, we couple two neural network architectures, namely skip-thought and self-organising map (SOM), to learn generic sentence representations that are particularly suited for assessing semantic similarity between sentences, and cluster these sentences together efficiently by topic. One key aspect of our approach compared to more traditional topic models such as LDA, is that at test time, when topic distributions need to be assigned to unseen documents, inference of the model's parameters does not need to be performed. In addition, we propose a novel human evaluation study to measure how well the inferred topics match human perception.

This chapter is organised as follows. We first present the skip-thought model and the SOM in Section 5.1. In Section 5.2, we present our method in detail and describe how documents are modelled as a bag-of-sentences. In Section 5.3, we empirically evaluate the performance our model in a classification task on a real-world dataset and report the results of our human study. We conclude in Section 5.4.

5.1 Preliminaries

Before detailing our approach, we first provide detail background on the skip-thought model in Section 5.1.1 and the SOM in Section 5.1.2 as they form the building blocks of our contriibution.

5.1.1 Skip-thought

A crucial first step in learning topics is the choosing an appropriate representation for sentences. As discussed in Section 2.1.2, the sentence embeddings approach has gained significant traction in recent years due to its ability to form clusters of similar meaning in the embedding space. One approach that provides high quality representation of sentences is skip-thought [Kiros et al., 2015]. Inspired by the skip-gram architecture [Mikolov et al., 2013] (Section 2.1.1), skip-thought learns distributed representations of sentences by using an encoder-decoder architecture to compute sentence embeddings. However, instead of using a word to predict its surrounding context, it uses sentences to predict the sentences around it. In particular, the encoder maps an input sentence to a vector representation, and two decoders are trained to minimise the error when the previous and next sentences are reconstructed from this vector (Figure 2.4). As a result, sentences that have similar syntax and semantics are likely to have similar vectors.

In more detail, assume a target input sentence $\mathbf{s}$ from a training set, its previous sentence $\overleftarrow{\mathbf{s}}$, and its next sentence $\overrightarrow{\mathbf{s}}$. Let $\mathbf{w}_t$, $\overleftarrow{\mathbf{w}_t}$ and $\overrightarrow{\mathbf{w}_t}$ be the one-hot encoded word vectors at time t for the target, previous and next sentences respectively such that $\mathbf{s} = \{\mathbf{w}_1, \cdots, \mathbf{w}_T\}$, $\overleftarrow{\mathbf{s}} = \{\overleftarrow{\mathbf{w}_1}, \cdots, \overleftarrow{\mathbf{w}_T}\}$, and $\overrightarrow{\mathbf{s}} = \{\overrightarrow{\mathbf{w}_1}, \cdots, \overrightarrow{\mathbf{w}_T}\}$. After appropriate mapping of the one-hot encoded words to embeddings through a projection layer, the encoder computes a hidden state vector $\mathbf{h}_T$ from the target sentence $\mathbf{s}$ using an LSTM[1] (Section 6.1.2). Then, conditioned on $\mathbf{h}_T$, forward and backward decoders are trained to reconstruct the sentences $\overleftarrow{\mathbf{s}}$ and $\overrightarrow{\mathbf{s}}$. That is, the initial state $\overrightarrow{\mathbf{h}}_0$ and $\overleftarrow{\mathbf{h}}_0$ of the LSTM of the forward and backward decoder respectively is $\mathbf{h}_T$. Both decoders are neural language models [Bengio et al., 2003] using an independent set of parameters. The training objective is to minimise the sum of log-probabilities of all the predicted words $\overleftarrow{\mathbf{w}_t}$ and $\overrightarrow{\mathbf{w}_t}$ for all t conditioned on $\mathbf{h}_T$. That is

$$\min \sum_t \log p\left(\overrightarrow{\mathbf{w}_t} | \overrightarrow{\mathbf{w}_{<t}}, \mathbf{h}_T\right) + \min \sum_t \log p\left(\overleftarrow{\mathbf{w}_t} | \overleftarrow{\mathbf{w}_{<t}}, \mathbf{h}_T\right), \tag{5.1}$$

where $\mathbf{w}_{<t} = \{\mathbf{w}_0, \cdots, \mathbf{w}_{t-1}\}$. This is an instance of multi-objective optimisation [Kalyanmoy, 2014] where several training targets are combined in one training scheme. It is worth noting that jointly minimising the sum

$$\min \left(\sum_t \log p\left(\overrightarrow{\mathbf{w}_t} | \overrightarrow{\mathbf{w}_{<t}}, \mathbf{h}_T\right) + \sum_t \log p\left(\overleftarrow{\mathbf{w}_t} | \overleftarrow{\mathbf{w}_{<t}}, \mathbf{h}_T\right) \right) \tag{5.2}$$

is bounded from below by Equation 5.1.

[1] We omit the deep recurrent architecture [Graves et al., 2013] from the original paper for simplicity of notation. When used, the hidden state of all layers at the last time step of the encoder are used as the initial state of all layers of the decoders.

Proposition 5.1. *If $min\, f(x)$, $min\, g(x)$, and $min\,(f+g)(x)$ are defined for all $x \in \mathbb{R}$, then*

$$min\,(f+g)(x) \geq min\, f(x) + min\, g(x)\,.$$

Proof. By definition, a is a minimum of $f(x)$ if $f(x) \geq a$ for all x, and there exists an x such that $f(x) = a$. Let b be the minimum of $g(x)$. It follows from the order axioms of real numbers that

$$f(x) \geq a \Rightarrow f(x) + g(x) \geq a + g(x)\,.$$

Therefore, since $g(x) \geq b$ we get

$$f(x) + g(x) \geq a + b \qquad \forall x. \tag{5.3}$$

Let c be the minimum of $(f+g)(x)$. There exist an x^* such that $x^* = \operatorname{argmin}_x (f+g)(x)$ and

$$c = (f+g)(x^*) \triangleq f(x^*) + g(x^*)\,.$$

Since Equation 5.3 holds for all x (including x^*), by replacing x by x^* in Equation 5.3 we obtain

$$c \geq a + b.$$

□

This means that searching the space of possible values for the model's parameters via stochastic gradient descent may result in better predictive performance when using Equation 5.1 rather than Equation 5.2. In the following section, we describe the second component used in our contribution enabling the clustering of sentence embeddings.

5.1.2 Self-organising Map

A self-organising map (SOM) [Kohonen, 1982] organises arbitrary vectors into groups without supervison. In our setting, this means that it can be used to cluster sentence embeddings into meaningful units. Fundamentally, a SOM (Section 2.1.3) is a dimensionality reduction algorithm that maps high-dimensional input vectors such as embeddings, to lower-dimensional representations (typically one or two dimensions) while preserving similarities. That is, inputs that are nearby in high-dimensional space will also be nearby in the low-dimensional space. It is a type of feedforward neural network in which the output nodes increasingly specialise to the extent that only one node gives a response to a given input. This type of competitive learning process [Ahalt et al., 1990] fundamentally differs to the more traditional backpropagation algorithm [Williams and Hinton, 1986] in that nodes have explicit meaning.

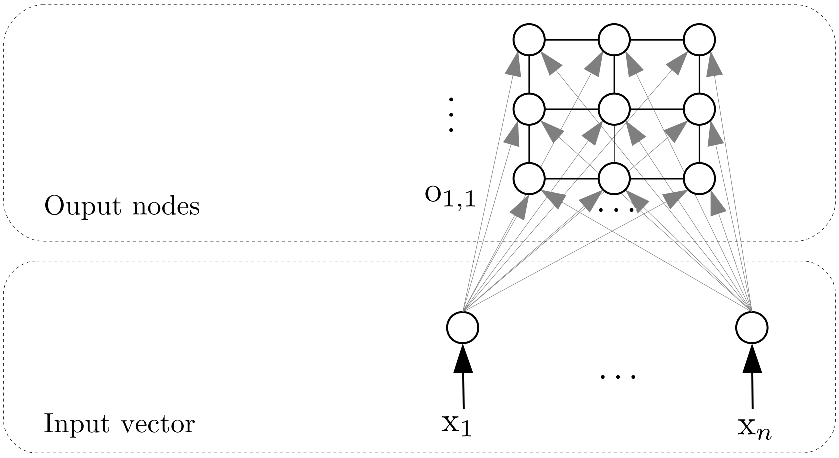


Figure 5.1: Architecture of a two dimensional SOM densely connecting an input vector $\mathbf{x}$ to the weight vectors $\mathbf{o}$.

In more detail, let the vector $\mathbf{x}_n$ be the n-th training sample (e.g. an embedding vector). For simplicity of presentation we assume a single dimensional output. Let $\boldsymbol{\theta}_{ij}$ be the weight between the input node i and the output node j. Furthermore, we denote by $\boldsymbol{\theta}_j = [\boldsymbol{\theta}_{1j}, \boldsymbol{\theta}_{2j}, \cdots]$ the vector of weights associated with the j-th output node. During training, each input sample $\mathbf{x}$ is presented sequentially to the SOM without specifying the desired output. The competitive training process first computes the Euclidean distance between the input $\mathbf{x}$ and the weight vector $\mathbf{w}_j$ associated with each output node j. The winning node c_n for input sample $\mathbf{x}_n$ is then defined as the output node j with the minimum distance to the input $\mathbf{x}_n$. That is

$$c_n = \arg\min_j ||\mathbf{x}_n - \boldsymbol{\theta}_j|| \,. \tag{5.4}$$

Finally, the weights are updated such that the winner node c_n (and its neighbours to a lesser extent) is brought closer to the input $\mathbf{x}_n$

$$\boldsymbol{\theta}_j \rightarrow \boldsymbol{\theta}_j + u_j \eta \left(\mathbf{x}_n - \boldsymbol{\theta}_j\right), \tag{5.5}$$

where $\eta \in \; [0, 1]$ is the learning rate and u_j is a neighbourhood function centered at c_n. The neighbourhood function u_j weighs the adjustments applied by each output node during training. Popular choices of neighbourhood functions include the Gaussian function

$$u_j = \exp\left(-\frac{|\mathbf{x} - c|_j^2}{2\pi\sigma\left(n\right)^2}\right), \tag{5.6}$$

and the Ricker wave

$$u_j = \left(1 - \frac{|\mathbf{x} - c|_j^2}{\pi\sigma\left(n\right)^2}\right) \exp\left(-\frac{|\mathbf{x} - c|_j^2}{2\pi\sigma\left(n\right)^2}\right).$$

The spread $\sigma\left(n\right)$ of the neighbourhood function u_j typically decays during training according to the current training step n. For example, the exponential decay function is given by

$$\sigma\left(n\right)=\sigma_0\text{exp}\left(-\lambda n\right) \tag{5.7}$$

for some decay constant λ. Initially, $\sigma\left(n\right)$ is large, forcing the neighbourhood function to include most output nodes on the map. As training proceeds (i.e. n increases) the neighbourhood shrinks down to just the winner node itself. This means that initially the update rule for each winner node has a very large field of influence which gradually shrinks to the point of influencing only the winner node. Training is completed when the map has converged or a maximum number of iterations has been reached. In measuring convergence, consideration must be taken for both the quantisation and the topological errors. On the one hand, the average quantisation error measures how well the map fits the input data. It is the average distance between each input vector $\mathbf{x}_n$, and the weight of its corresponding winning node $\mathbf{w}_{c_n}$

$$E_q=\frac{1}{N}\sum_n\left|\left|\mathbf{x}_n-\mathbf{w}_{c_n}\right|\right|,$$

where N is the size of the training set. On the other hand, the average topological error measures how well the topology is preserved. Unlike the average quantisation error, it considers the structure of the map. It is given by

$$E_t=\frac{1}{N}\sum_n h\left(\mathbf{x}_n\right)$$

where $h\left(\mathbf{x}_n\right)$ is 1 if the first and second winner nodes of $\mathbf{x}_n$ are adjacent, 0 otherwise. The average topographic error increases if the winner node and the second winner node of an input vector are adjacent to each other on the map. After training, when a new input $\mathbf{x}$ is presented, the corresponding output c is found based on the minimum distance rule (Equation 5.4). It should be noted that the size and dimensionality of the output, the neighbourhood function and the learning rate are generally selected on the basis of heuristic information [Grieco et al., 2017]. Furthermore, although two dimensional outputs are often chosen, there are no theoretical limits on using higher or lower dimensions. However there may be computational limits, and higher dimensional outputs may be more difficult to interpret. The training process of the SOM is summarised in Procedure 8.

In isolation, the skip-thought and the SOM do not address Aim 2. The skip-thought neither score similarity between sentences, nor their importance in a topic. Similarly, the SOM does not learn useful sentence features that would enable a grouping per topic. In the next section, we introduce our contribution which combines the benefits of both architectures and allow the learning of sentence-based topics, where the sentences are ranked by their importance in each topic.

Algorithm 8 Training of the SOM.

```
Input: a hiden state vector $\mathbf{h}$,
Parameters: nb of output nodes $K$, neighbourhood function $f$ and its spread $\sigma_0$, learning rate $\eta_0$, decay
function $g$, max nb of epochs $N$.

$\boldsymbol{\theta}_i \sim \text{Uni}([0,1))$                    ▷ Initialise the weights of the entire network for all $i$

$\mathbf{h} = \frac{\mathbf{h}}{|\mathbf{h}|}$                    ▷ Normalise the hidden state $\mathbf{h}$
$\boldsymbol{\theta}_i = \frac{\boldsymbol{\theta}_i}{|\boldsymbol{\theta}_i|} \forall i \in \{0, \cdots, K\}$                    ▷ Normalise the weight vectors

for each iteration $n \in \{0, \cdots, N\}$ do
    for each $x \in \{0, \cdots, K\}$ do
        $\mathbf{o}_j = |\mathbf{h} - \boldsymbol{\theta}_j|$                    ▷ Forward propagate $\mathbf{h}$
    end for

    $c = \text{argmin}_j(\mathbf{o})$                    ▷ Find winner node

    $\sigma = g(\sigma_0, n)$                    ▷ Update neighbourhood function spread
    $\eta = g(\eta_0, n)$                    ▷ Update learning rate
    for each $x \in \{0, \cdots, X\}$ do
        $\boldsymbol{\theta}_x = \mathbf{w}_x + \eta \times f(x, c, \sigma) \times (\mathbf{h} - \boldsymbol{\theta}_x)$                    ▷ Update weight vector
        $\boldsymbol{\theta}_i = \frac{\boldsymbol{\theta}_i}{|\boldsymbol{\theta}_i|}$                    ▷ Normalise weight vector
    end for

end for

return $c$
```

5.2 Clustering Sentences per Topic

We address the challenge of learning sentence-based topics, along with the mixture distribution over these topics for each document, by coupling a skip-thought with a SOM. While the skip-thought computes a high-dimensional embedding for each input sentence, the SOM performs a non-linear mapping between these embeddings and a two-dimensional lattice of output nodes. The dimensionality reduction performed by the SOM enables the grouping of sentences with similar topics together by mapping related sentences to the same output node, or at worst, one of its closest neighbour.

In more detail, at each training iteration, an input sentence $\mathbf{s}$ is fed to the encoder of the skip-thought, producing a hidden state $\mathbf{h}_T$ at the last time-step T. This hidden state $\mathbf{h}_T$ represents the embedding for the input sentence $\mathbf{s}$. The value of the hidden state $\mathbf{h}_T$ is used simultaneously as initial state of the two decoders. As with the original skip-thought, we penalise the decoders using the sum of negative log-likelihood loss function at each time-step i (Equation 5.1). More importantly, the value of the hidden state $\mathbf{h}_T$ is also used as input to the SOM (Figure 5.2). In order for each individual component of $\mathbf{h}_T$ to contribute proportionately when presented to the SOM, we normalise $\mathbf{h}_T$ such that $\sum_k \mathbf{h}_{T,k} = 1$. This avoids large values of the individual component $\mathbf{h}_{T,i}$ skewing the distance between the embeddings and the output nodes. In turn, we also normalise the weights of the SOM after each iteration. Furthermore, we assume a univariate Gaussian neighbourhood function f and an exponential decay function g.

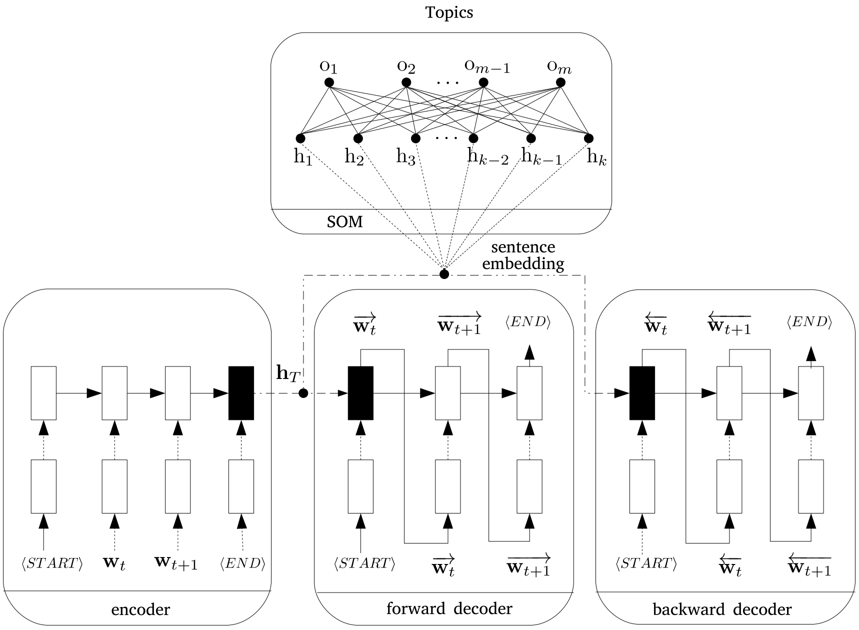


Figure 5.2: Coupling of the SOM's input to the skip-though's hidden state.

For each document a distribution over these clusters (i.e. group of sentences) is built. To do so, we associate a unique document identifier for each sentence, such that each sentence in a document has the same unique identifiers. Once the model is trained, we present all instances from the test set to the network and build a histogram over the winning output nodes for each document. Candidate sentences are then ranked according to quantisation error, and the k-top ranked sentences are chosen to represent the topics.

When training is over, the two decoders are discarded since since they do not contribute to the feedforward activation within the network. Therefore, we are left with the encoder and the SOM.

5.3 Experimental Evaluation

We evaluate the performance of our model on document classification and human evaluation. Since our objective is to demonstrate that sentences are easier to interpret for humans when it comes to topics (Aim 2), we benchmark our assumption against a unigram-based model. Specifically, we chose LDA (Section 2.2) as benchmark as it is the state-of-the-art and most widely used unigram topic model. In particular, we use the highly optimised implementation of LDA provided by the Gensim[2] library.

[2] https://radimrehurek.com/gensim/

5.3.1 Dataset

Our model, and in particular its skip-thought component, requires a large corpus of semantically contiguous sentences for training[3]. Books fulfill this requirement while also being a rich source of topics. As such, the BookCorpus dataset[4] [Zhu et al., 2015] is a natural choice given its size and diversity of genres. It is a collection of 11,038 books written in English categorised into 16 different genres collected from Smashwords[5] – an e-book distribution platform for independent authors and publishers. After the removal of empty books, books with unknown character encodings format, duplicated books across genres, and books consisting of the concatenation of other books, we are left with 4,184 usable candidates (38% of the total number of books) totaling about 9.2 million sentences to be processed by our model and benchmark (Table 5.1).

Adventure	Fantasy	Historical	Horror
223	352	117	222
Humor	Literature	Mystery	New adult
174	253	245	40
Young adult	Romance	Science fiction	Teen
117	1,199	358	275
Themes	Thriller	Vampires	Other
25	314	110	160

Table 5.1: Number of books per genre in the BookCorpus dataset after the exclusion of undesirable books.

The average number of sentences per book across genres is 2,200±1,900 revealing a wide standard deviation in books' length (Figure 5.3).

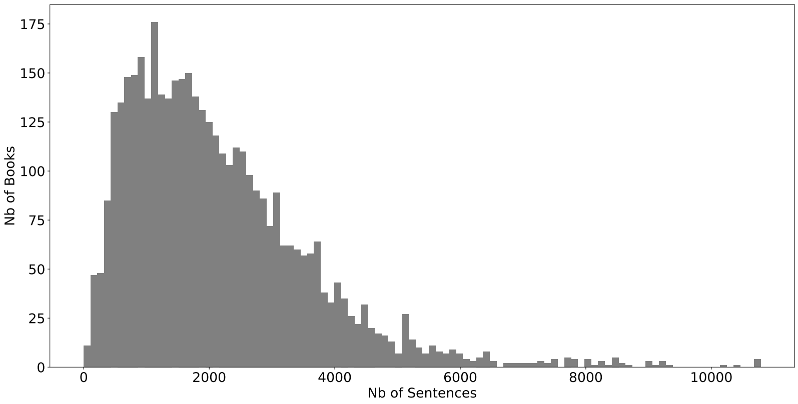


Figure 5.3: Distribution of the number of sentences per book.

The average length of a sentence is 15±16.38 words (Figure 5.4) which is a reasonable range for recurrent architectures such as LSTM to perform well [Bahdanau et al., 2014].

[3] This precondition is no longer required in actual use since the decoders are discarded.

[4] http://yknzhu.wixsite.com/mbweb

[5] smashwords.com

Furthermore, in view of the human evaluation (Section 5.3.4), longer sentences may be harder to hold in memory and are more likely to be unclear (i.e. harder to parse and understand or have too many details or information) [Pinker, 2014].

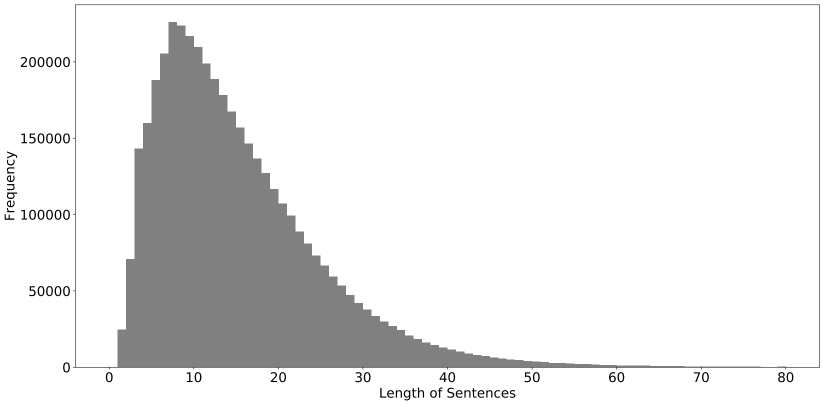


Figure 5.4: Distribution of sentence's length across the corpus.

5.3.2 Training Details

The parameters and hyperparameters of each component are set as follows.

Skip-thought. We initialise the weights uniformly between 0 and 1. We remove sentences with more than 25 words (i.e. the mean sentence length plus one standard deviation), and unroll the LSTMs for the equivalent number of timesteps. To enable training with mini-batch stochastic gradient descent, we pad sentences with less than 25 words with zero vectors. Furthermore, we reverse the word order in the input sentence to improve the LSTM performance [Sutskever et al., 2014]. We set the size of words embedding vectors to 620 and the size of the hidden state of the LSTM to 2,400. All hyperparameters are shared between the encoder and the two encoders. We use the ADAM optimiser [Kingma and Ba, 2014] with learning rate 8×10^{-4}.

SOM. We initialise the weights with random values in the range $[0, 1]$ while ensuring that each weight vector $\boldsymbol{\theta}_j$ has a norm of one. We use a two-dimensional output size and an input dimension of 2,400, that is, the size of the LSTM hidden state vector. We elect a Gaussian neighbourhood function (Equation 5.6) parameterised by an exponential decay function (Equation 5.7) with decay constant $\lambda = 1$. We set the learning rate to $\eta = 0.5$.

5.3.3 Document Classification

In this section, we conduct a binary classification experiment to categorise books into two genres. We detail our experimental setting in Section 5.3.3.1. We discuss the results in Section 5.3.3.2.

5.3.3.1 Experimental Setting

We select two genres at random and sample exactly 200 books in each genre. We further constraint each book to have exactly 1,000 sentences with a vocabulary size of 400 words. We split the books into a training (70%), a validation (25%), and a testing sets (5%) ensuring that each split has an equal number of books in each genre (i.e. the test set has exactly 10 books per genre). We train a support vector machine (SVM) classifier [Cortes and Vapnik, 1995] using the topic mixture distributions as features for both our model and LDA. We assess the performance of the classifier using the accuracy measure (Section 6.4.2). In addition to LDA, we include a new baseline which assigns a genre at random to each document with 50% probability. Finally, we run each model 150 times with different initial values to ensure statistically significant results at the 99% confidence level.

5.3.3.2 Results

In this experiment, we are interested in measuring the classification accuracy as a function of the number of topics (Figure 5.5). We observe that increasing the number of topics improves the classification accuracy for both our model and LDA up to certain threshold (around 80 topics in this case). As can be seen, LDA performs exceptionally well on this task with an accuracy of 90% against 70% for our approach. This can be explained by a number of facts. First, our model has a much larger number of parameters (i.e. around 70 million) compared to LDA (i.e. 40,100 for 100 topics)[6]. The number of parameters imposes a lower bound on the number of training examples required. Therefore, training our model on a dataset that is smaller than the number of parameters greatly affects its generalisation performance. Second, LDA has far fewer hyperparameters to tune manually. We expect that the appropriate tuning of our model's hyperparameters is likely to largely improve on performance.

[6] We note that the number of parameters of our model is independent of the size of the vocabulary.

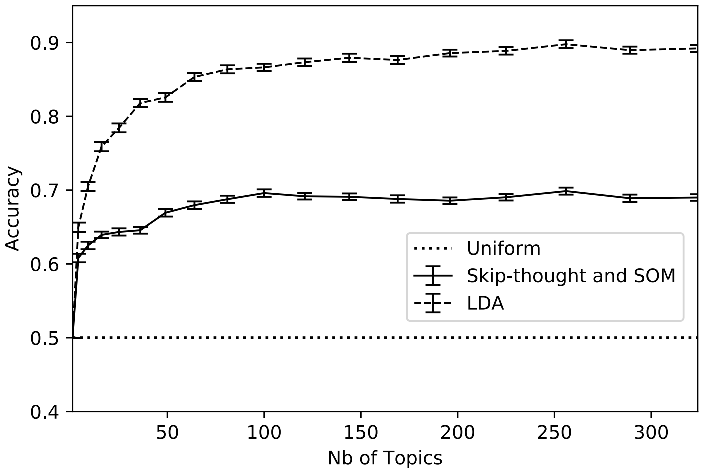


Figure 5.5: Accuracy on binary classification from the BookCorpus dataset when increasing the number of topics.

Given this, further insights can be gleaned into the behaviour of our model by looking at the SOM's parameters and activation maps. First, we observe from the parameters map (Figure 5.6) that the classification accuracy achieved is dependent on both the decay constant σ and the learning rate η. However, it is not clear from the map which combination of parameters should be used to achieve the best accuracy since many combinations result in the same value.

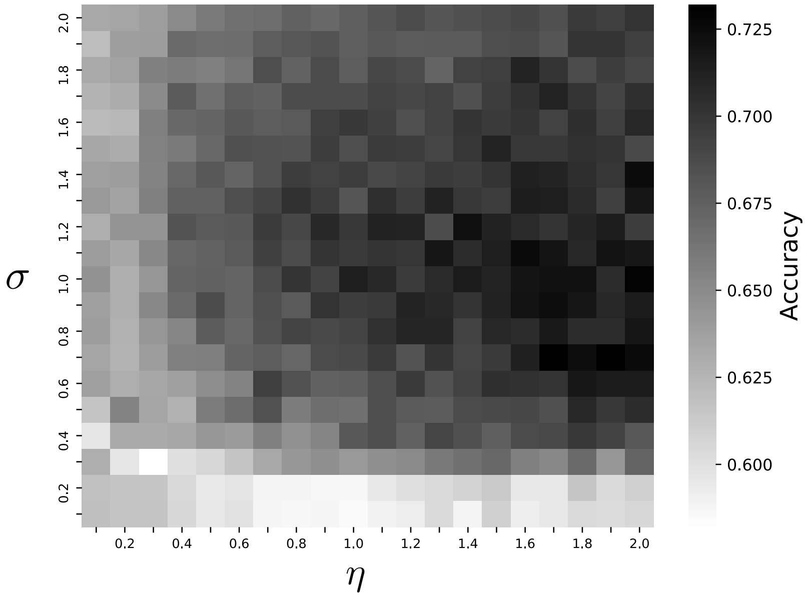


Figure 5.6: Classification accuracy against the decay constant σ and learning rate η (100 topics).

Figure 5.7 shows the final activation patterns after training with 100 topics. The clusters of topics are easily identifiable and densely packed in the lower right-hand side of the two-dimensional map. We further observe a number of nodes with no sentences assigned. Such *dead nodes* result from weight values initialised far from the input vectors in the training set. It is the case that reducing the number of dead nodes leads to more informative feature vectors, thus more accurate classification. A simple way to address this is to initialise the weights of the SOM by picking input vectors at random from the training set. We leave this enhancement for future work.

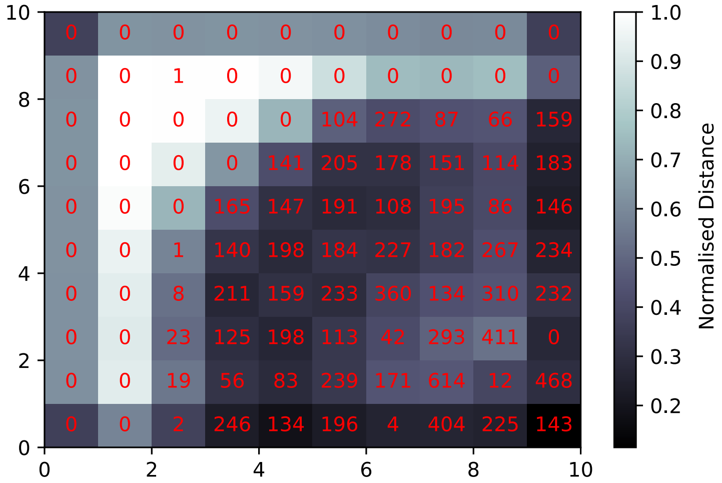


Figure 5.7: Activation map of a 10 × 10 grid (i.e. 100 topics) after training. The numbers in red indicate the total of number of sentences assigned to a particular node. The distance map is visualised in the background, where the lighter shades of grey indicate low normalised sum of the distances between a node and its neighbours.

5.3.4 Human Evaluation

In this study, we assess via human judgments the capability of our model to assign unambiguous topics to a subset of documents from the BookCorpus dataset compared to the benchmark LDA. In particular, we test the following hypothesis:

> Thematic relations are easier to infer when presented as a list of sentences rather than a list of words.

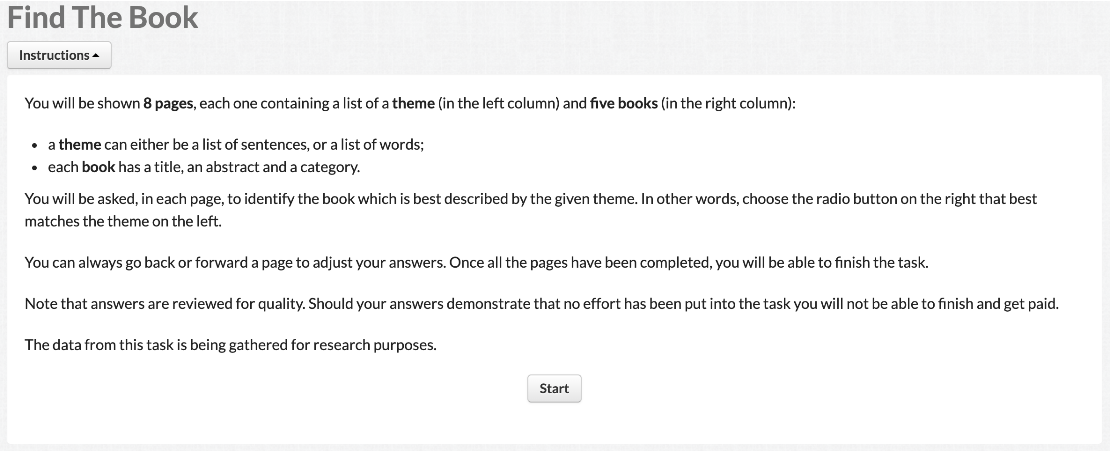
We detail the design of the study in Section 5.3.4.1. We introduce the performance metrics used in Section 5.3.4.2. We discuss the results in Section 5.3.4.3.

5.3.4.1 Experiment Design

The experiment is conducted on the Figure Eight crowdsourcing platform[7] (formerly CrowdFlower). A number of annotators were recruited and asked to select the single

[7] www.figure-eight.com/

book among three alternatives that best matches a given topic (Figure 5.8). To support the annotators in their decision, they were shown the title and category of each book. Furthermore, since we did not expect the annotators to be familiar with the books prior to the experiment, we also showed the books' abstract that were extracted manually from a number of websites (e.g. Amazon[8] or Goodreads[9]).



Find The Book

You will be shown **8 pages**, each one containing a list of a **theme** (in the left column) and **five books** (in the right column):

- a **theme** can either be a list of sentences, or a list of words;
- each **book** has a title, an abstract and a category.

You will be asked, in each page, to identify the book which is best described by the given theme. In other words, choose the radio button on the right that best matches the theme on the left.

You can always go back or forward a page to adjust your answers. Once all the pages have been completed, you will be able to finish the task.

Note that answers are reviewed for quality. Should your answers demonstrate that no effort has been put into the task you will not be able to finish and get paid.

The data from this task is being gathered for research purposes.

Figure 5.8: Screenshot of the description page of a HIT (i.e. a group of eight tasks) as presented to the annotators.

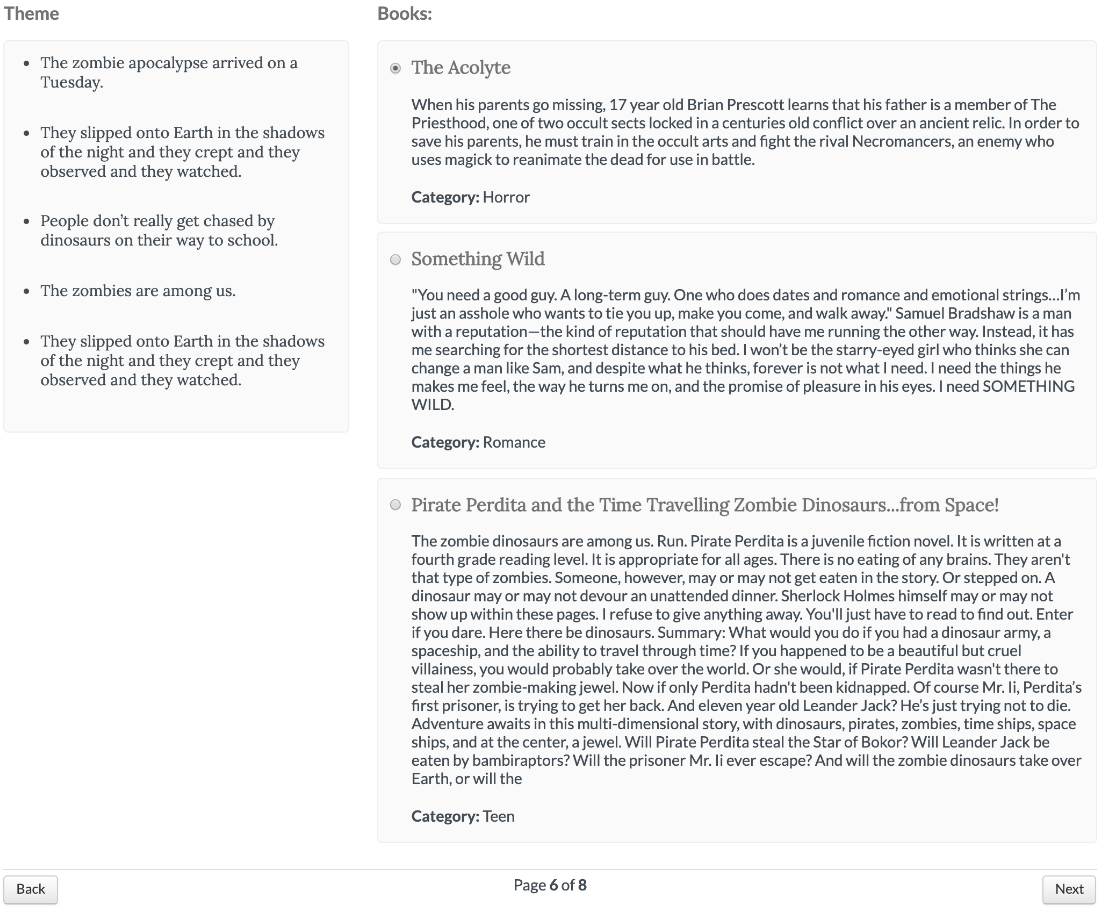
To construct a task (Figure 5.9 and Figure 5.10), we first selected a topic at random that has been identified by either our model or LDA from the test set. We then selected the five highest-ranking sentences (for our model) or words (for LDA) from that topic under the heading "theme"[10]. Furthermore, we selected three books per task from the test set across all genres. The three books consisted of the highest-ranking, mid-ranking, and lowest-ranking books assigned to that topic. The ground truth for each task was the highest-ranking book assigned to the topic. All three books were shuffled before being presented to the annotators to avoid biases in their responses. We organised the tasks into groups of eight to be completed as a single unit by each annotator. We refer to these groups as human intelligence tasks (HITs). Participants were prevented from participating in more than one HIT. Therefore the same task was repeated multiple time across different HITs (but not within the same HIT) to obtain multiple judgments for the same task from different annotators. Out of the eight tasks that comprises a HIT, half consisted of topics made up of sentences (as identified by our model), and the other half of the topics made up of words (as identified by LDA). Within a HIT, annotators were required to complete each group before moving on to the second group. The order in which these groups appeared to the annotators were shuffled to avoid potential ordering effects and first sample bias [Moskowitz, 1977]. Furthermore, participants were screened for quality using a *honey pot task* in each group within a HIT. Such tasks have known

[8] www.amazon.co.uk

[9] www.goodreads.com/

[10] We referred to a "topic" as a "theme" in each task to avoid ambiguity.

answers and are very easy to complete. Participants who failed on these tasks were not suitable and their answers were discarded. In total, the number of honey pot tasks amounted to two per HIT. The location of these honey pot tasks within each group was shuffled as to avoid predictability. Participants did not have any time limit to complete a HIT. Finally, upon completion of the HITs, participants were asked to complete an optional satisfaction survey, rating a number of factors (overall satisfaction, clarity of instructions, simplicity of the task, fairness, and payment) on a scale between 0 and 5.



Theme

- The zombie apocalypse arrived on a Tuesday.
- They slipped onto Earth in the shadows of the night and they crept and they observed and they watched.
- People don't really get chased by dinosaurs on their way to school.
- The zombies are among us.
- They slipped onto Earth in the shadows of the night and they crept and they observed and they watched.

Books:

● The Acolyte

When his parents go missing, 17 year old Brian Prescott learns that his father is a member of The Priesthood, one of two occult sects locked in a centuries old conflict over an ancient relic. In order to save his parents, he must train in the occult arts and fight the rival Necromancers, an enemy who uses magick to reanimate the dead for use in battle.

Category: Horror

○ Something Wild

"You need a good guy. A long-term guy. One who does dates and romance and emotional strings...I'm just an asshole who wants to tie you up, make you come, and walk away." Samuel Bradshaw is a man with a reputation—the kind of reputation that should have me running the other way. Instead, it has me searching for the shortest distance to his bed. I won't be the starry-eyed girl who thinks she can change a man like Sam, and despite what he thinks, forever is not what I need. I need the things he makes me feel, the way he turns me on, and the promise of pleasure in his eyes. I need SOMETHING WILD.

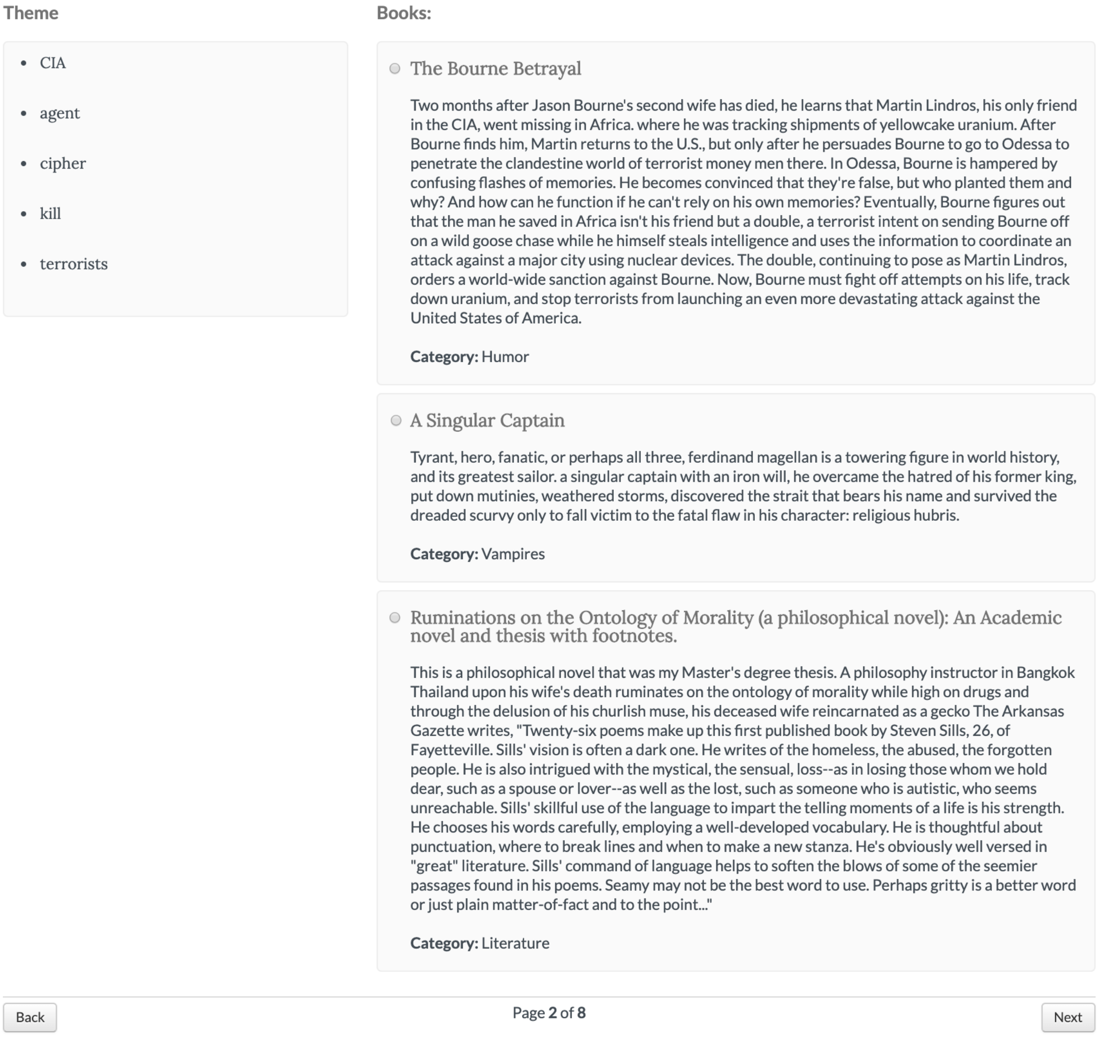
Category: Romance

○ Pirate Perdita and the Time Travelling Zombie Dinosaurs...from Space!

The zombie dinosaurs are among us. Run. Pirate Perdita is a juvenile fiction novel. It is written at a fourth grade reading level. It is appropriate for all ages. There is no eating of any brains. They aren't that type of zombies. Someone, however, may or may not get eaten in the story. Or stepped on. A dinosaur may or may not devour an unattended dinner. Sherlock Holmes himself may or may not show up within these pages. I refuse to give anything away. You'll just have to read to find out. Enter if you dare. Here there be dinosaurs. Summary: What would you do if you had a dinosaur army, a spaceship, and the ability to travel through time? If you happened to be a beautiful but cruel villainess, you would probably take over the world. Or she would, if Pirate Perdita wasn't there to steal her zombie-making jewel. Now if only Perdita hadn't been kidnapped. Of course Mr. Ii, Perdita's first prisoner, is trying to get her back. And eleven year old Leander Jack? He's just trying not to die. Adventure awaits in this multi-dimensional story, with dinosaurs, pirates, zombies, time ships, space ships, and at the center, a jewel. Will Pirate Perdita steal the Star of Bokor? Will Leander Jack be eaten by bambiraptors? Will the prisoner Mr. Ii ever escape? And will the zombie dinosaurs take over Earth, or will the

Category: Teen

Figure 5.9: Screenshot of a task designed to assess our model. Annotators were given three books' title, abstract and categories (right-hand side). The annotators must select which of the three books is best described by the set of sentences in the "theme" column (left-hand side).



Theme

- CIA
- agent
- cipher
- kill
- terrorists

Books:

○ The Bourne Betrayal

Two months after Jason Bourne's second wife has died, he learns that Martin Lindros, his only friend in the CIA, went missing in Africa. where he was tracking shipments of yellowcake uranium. After Bourne finds him, Martin returns to the U.S., but only after he persuades Bourne to go to Odessa to penetrate the clandestine world of terrorist money men there. In Odessa, Bourne is hampered by confusing flashes of memories. He becomes convinced that they're false, but who planted them and why? And how can he function if he can't rely on his own memories? Eventually, Bourne figures out that the man he saved in Africa isn't his friend but a double, a terrorist intent on sending Bourne off on a wild goose chase while he himself steals intelligence and uses the information to coordinate an attack against a major city using nuclear devices. The double, continuing to pose as Martin Lindros, orders a world-wide sanction against Bourne. Now, Bourne must fight off attempts on his life, track down uranium, and stop terrorists from launching an even more devastating attack against the United States of America.

Category: Humor

○ A Singular Captain

Tyrant, hero, fanatic, or perhaps all three, ferdinand magellan is a towering figure in world history, and its greatest sailor. a singular captain with an iron will, he overcame the hatred of his former king, put down mutinies, weathered storms, discovered the strait that bears his name and survived the dreaded scurvy only to fall victim to the fatal flaw in his character: religious hubris.

Category: Vampires

○ Ruminations on the Ontology of Morality (a philosophical novel): An Academic novel and thesis with footnotes.

This is a philosophical novel that was my Master's degree thesis. A philosophy instructor in Bangkok Thailand upon his wife's death ruminates on the ontology of morality while high on drugs and through the delusion of his churlish muse, his deceased wife reincarnated as a gecko The Arkansas Gazette writes, "Twenty-six poems make up this first published book by Steven Sills, 26, of Fayetteville. Sills' vision is often a dark one. He writes of the homeless, the abused, the forgotten people. He is also intrigued with the mystical, the sensual, loss--as in losing those whom we hold dear, such as a spouse or lover--as well as the lost, such as someone who is autistic, who seems unreachable. Sills' skillful use of the language to impart the telling moments of a life is his strength. He chooses his words carefully, employing a well-developed vocabulary. He is thoughtful about punctuation, where to break lines and when to make a new stanza. He's obviously well versed in "great" literature. Sills' command of language helps to soften the blows of some of the seemier passages found in his poems. Seamy may not be the best word to use. Perhaps gritty is a better word or just plain matter-of-fact and to the point..."

Category: Literature

Figure 5.10: Screenshot of a task designed to assess LDA. Annotators were presented with a theme comprised of a set of words extracted from LDA (left-hand side), and were asked to select the book which best match the theme based the books' title abstract and category (right-hand side).

5.3.4.2 Performance Metrics

We use two performance metrics to assess which of our model or the benchmark is best suited for the tasks in this study: a standard classification measure and its outliers-adjusted variation.

Accuracy. The accuracy measures the ratio of correctly identified books, normalised by the total number of judgments in each task. Following our notation from Equation 3.12, the accuracy is given by

$$\text{Accuracy} = \frac{\sum_i \pi_{i,i}}{\sum_{i,j} \pi_{i,j}}, \tag{5.8}$$

where $\sum_i \pi_{i,i}$ is the total number of correct judgments, and $\sum_{i,j} \pi_{i,j}$ the total number of judgments. Table 5.2 illustrates the steps for determining the accuracy in a HIT consisting of two tasks.

	Judgments	Ground truth	$\sum_i \pi_{i,i}$	$\sum_{i,j} \pi_{i,j}$
Task 1	1,1,3,1	1	3	4
Task 2	2,3,2,3	3	2	4

Table 5.2: Example of two tasks with four judgments each.

It follows that the accuracy for this example is $\frac{3+2}{4+4} = 62.5\%$.

Plurality voting (PV). For each task, plurality voting selects the book that has received the largest number of judgments if the solution is unique (Equation 3.3). In case of ties, a book is randomly selected among the most likely candidates. Once the PV for each task is obtained, the accuracy is used (Equation 5.8) to compute the final scores. Building on the prior example (Table 5.2), the PV for the two tasks are

Task 1: $\mathrm{PV}_1 = 1,$

Task 2: $\mathrm{PV}_2 = 2$ or $\mathrm{PV}_2 = 3$, depending on the random algorithm used to break ties.

Applying the accuracy to the PV values, the final PV score is either $\frac{1}{1+1} = 50\%$ if $\mathrm{PV}_2 = 2$, or $\frac{1+1}{1+1} = 100\%$ if $\mathrm{PV}_2 = 3$. Plurality voting is less sensitive to outliers within the same task. This is because plurality voting uses the mode of the vote distribution in each task (Equation 3.3). However, we do not expect a large difference between the two metrics as the number of alternatives is small (i.e. three books in each task).

5.3.4.3 Results

We considered 15 topics[11] per model, each of which receives 10 judgments for a total of 300 judgments collected. These requirements led to the creation of 50 HITs[12] for which 50 participants were recruited. Individual HITs were remunerated 15 cents for a total of \$7.5 paid to the participants.[13]

The results of our experiment are shown in Table 5.3. We observe that LDA achieves better performance overall on this set of tasks based on the two metrics used.

[11] This number does not include the the additional topic used for the honey pot task.

[12] The number of HITs is computed as the number of topics, times the number of judgments per topic, times the number of models, divided by the number of tasks per HIT, minus the number of honey pot tasks. That is, $\frac{10\times15\times2}{8-2} = 50$.

[13] In practice, Figure Eight charges a 20% commision, increasing the total amount paid to \$9.

	Our model	LDA
TPR	21.33% (32/150)	**64.67%** (97/150)
PV	26.67% (4/15)	**73.33%** (11/15)

Table 5.3: Results of the comparison between our model and LDA using the TRP and PV performance metrics. The best performing model is highlighted in bold.

Although we anticipated a favourable, or at least comparable outcomes, these results confirm that while LDA has a number of drawbacks, it remains a very strong baseline. This forces us to reassess the initial assumptions behind our model. In particular, the assumption that the top-k semantic units within a topic describe one coherent theme may not be true for sentences as it is true for words alone. In contrast to LDA (which ranks the words by probability of co-occurrence), our approach clusters and ranks the sentences by syntactic and semantic similarity which is a different concept. Furthermore, we observed that most of the top-k sentences forming a topic were generic, and did not allow a precise identification of a particular book. For instance, the sentences forming one of the topics for a book in the young adult genre consist of: “I climb out onto the perch and look down”, “I run to the oven and turn it off, grab my phone and dial John”, and “I turn to watch Tucker draw from the jar”. It is clear that this particular set of sentences could be related to a wide variety of books in any genres. Although such generic topics are desirable when performing extractive multi-document summary (EMDS) (Chapter 1) where summaries should only contain information that is relevant to the topic of the entire corpus, they make the choice of a specific book more challenging.

Given this, our study was well designed and clearly achieved the objective it was intended for. This fact is supported by the positive feedback received by the participants. Out the 50 participants, 22 took part took part in the optional satisfaction survey. The participants were satisfied with the HIT overall (rated 3.8/5), the clarity of the instructions (rated 3.6/5), simplicity of the task (3.5/5), and the payment (rated 4.2/5).

5.4 Summary

In this chapter, we introduced a novel approach to learn topics from a corpus of unstructured textual documents. One of the key innovations of our method is the clustering and ranking of sentences by topics with the objective of increasing interpretability compared to unigram models. Another key aspect of our approach is the fact that no re-training of the model's parameters is required when performing inference on new documents. We demonstrated that our relatively unoptimised approach is a strong new baseline for text classification. As such, it is likely the case that additional fine-tuning would lead to further improvements. Furthermore, the design of evaluation frameworks capturing common practical usage of topic models remains an important open research direction [Blei, 2012]. Our human evaluation framework is a step forward toward validating document modeling assumptions for specific tasks (e.g. corpus exploration or information retrieval).

We have identified a number of future directions that may further improve performance:

- **Questionnaire type.** It has been observed that the choice of rating scales has an impact on rankings [Turpin et al., 2015]. Although our use of binary scale questions are a popular and an easy to set up type of questionnaire, they do not allow participants to submit nuanced answers. Rating scales such as Likert-scale [Likert, 1932] (e.g. strongly agree, agree, neutral, disagree, strongly disagree), or magnitude estimation [Stevens, 1975] (i.e. an unbounded or bounded real number) among others, allow finer-grained estimations of the strength of the relationship between the topics and particular books. This would provide additional insight into the annotators' perception of topic relevance.

- **Joint training.** Training our model sequentially, as we did, prevents the SOM from leveraging knowledge about the original data distribution (i.e. the input of the skip-thought), which in turn leads to a suboptimal performance. Hence, if the skip-thought does not represent the input data sufficiently well, the SOM may learn from imperfect data. Our original intent was to train the skip-thought and the SOM jointly. One approach consists of training the skip-though and the SOM iteratively. That is, at each training iteration, first update the weight of the skip-thought independently from the SOM, then train the SOM using the sentence embeddings computed from the current state of the skip-thought. We abandoned this idea, however, for the reason that such procedure breaks the assumption of stationarity since the input distribution of the SOM changes as the skip-thought is being trained; a phenomenon known as covariate[14] shift[15] [Sugiyama and Kawanabe, 2012, Ioffe and Szegedy, 2015]. Another approach consists of designing a global optimisation objective to minimise both the quantisation error of the SOM and the backpropagation error of the skip-thought simultaneously [Weijters et al., 1997, Weijters, 1995, Goppert and Rosenstiel, 1993].

[14] The term covariate denotes input [Quinonero Candela et al., 2008].

[15] The concept of covariate shift can be applied to parts of a learning model rather than the whole [Ioffe and Szegedy, 2015].

Chapter 6

Topic Identification and Tracking in Dialogues

In the previous chapter, we introduced a new approach to learning sentence-based topics with the objective of providing more context when assessing the semantics of each topic than unigram-based topic models. In this chapter, we address Aim 3 and propose a new method that for the first time simultaneously identifies the topic of each utterance in a dialogue session when such topics are both restricted, and not restricted to a limited set of alternatives. We established in Section 2.3.3 that handling both of these settings is important to ensure efficiency and accuracy in the response generated by the downstream dialogue management system. To make the distinction clear, we define a *topic class* as a small finite set of topic identifiers that have to be known a priori (e.g. HOTEL or RESTAURANT). And we define a *topic* as a phrase describing the finer-grained theme of an utterance (e.g. "flea market" or "education in the UK"). A key aspect of our approach is the use of external knowledge from Wikipedia articles[1] (i.e. title and content) as a reference point for identifying the topic of each utterance. In particular, the mapping of each Wikipedia article in a semantic space allows new and unseen utterances to be mapped to the closest article in terms of semantic similarity chosen from a vast pool of potential candidates.

We first give a detailed description of the key components of our proposed architecture in Section 6.1. We then formally introduce our contribution in Section 6.2. We give a step-by-step example in Section 6.3 to illustrate the process of inferring and evaluating the topics and the topic classes. We subsequently evaluate our model on a real-world dataset in Section 6.4. Finally, we conclude in Section 6.5.

[1] en.wikipedia.org

6.1 Preliminaries

Our proposed model is an extension of Kim et al. [2016] (Section 2.3.3). In particular, the authors propose an architecture based on a composition of a convolutional neural network (CNN) [LeCun et al., 1999] and a long-short term memory network (LSTM) [Hochreiter and Schmidhuber, 1997] commonly refered to as long recurrent convolutional neural network (LRCN) [Donahue et al., 2017, Karpathy and Fei-Fei, 2015, Vougiouklis et al., 2016]. In this section, we provide background details of the Kim et al. model prior to describing our extension in Section 6.2. More specifically, we first present the CNN architecture in Section 6.1.1, and then introduce the LSTM and the combined LRCN in Section 6.1.2.

6.1.1 Topic Classification from a Single Utterance

Following Kim [2014], utterances can be classified into topic classes using CNNs. Although CNNs have traditionally been used in computer vision to classify images, they have more recently been shown to also be effective in classifying text [Collobert et al., 2011, Kim, 2014]. In more detail, a CNN is a type of multi-layer feedforward neural network which consists of an input layer, an output layer, and a number of hidden layers (Figure 6.1). The hidden layers hierarchically learn more complex patterns from smaller ones in the input data (i.e. the utterances). A distinctive characteristic of CNNs is that they learn feature vectors that would be hand-engineered in traditional algorithms. More precisely, the CNN takes as input an utterance $\mathbf{u} = \{\mathbf{w}_1, \cdots, \mathbf{w}_N\}$ where N is the number of words in the utterance. An utterance may be comprised of one or multiple sentences. Each element $\mathbf{w}_i \in [0, |V|]$ is the index of a word in the vocabulary. A projection layer then maps each discrete word index to a continuous embedding vector resulting in a matrix $\mathbf{U} \in \mathbb{R}^{N \times K}$ where K is the size of the embedding vector. The CNN then performs a one-dimensional convolution over $\mathbf{U}$ by sliding a set of R two-dimensional filters of given height M_i and fixed width K (the same as the width of the input) over the rows of $\mathbf{U}$. Each filter is applied to $\mathbf{U}$ to generate a feature map $\mathbf{v}$. Each feature element $\mathbf{v}_i \in \mathbb{R}$ of a feature map is generated from a subregion $\mathbf{U}_{i:i+M-1}$ of the input utterance from the i-th to the $(i + M - 1)$-th row such that

$$\mathbf{v}_i = f\left(\Theta \odot \mathbf{U}_{i:i+M-1} + \mathbf{b}\right)$$

where f is an activation function (e.g. ReLU or sigmoid) and the operator $\odot$ performs an element-wise multiplication. The weights $\Theta \in \mathbb{R}^{M \times K}$ and biases $\mathbf{b} \in \mathbb{R}^{M}$ of each filter are shared for all i. Furthermore, the weights Θ and biases $\mathbf{b}$ of each filter are randomly initialised so that filters of the same size converge to different values after training. To capture the salient feature of each feature map, a max-pooling is applied resulting in a feature vector for each filter. Finally, the feature vector of each filter are concatenated into a single feature vector **x** and feed to a fully connected layer and a softmax performing the classification over the topic classes.

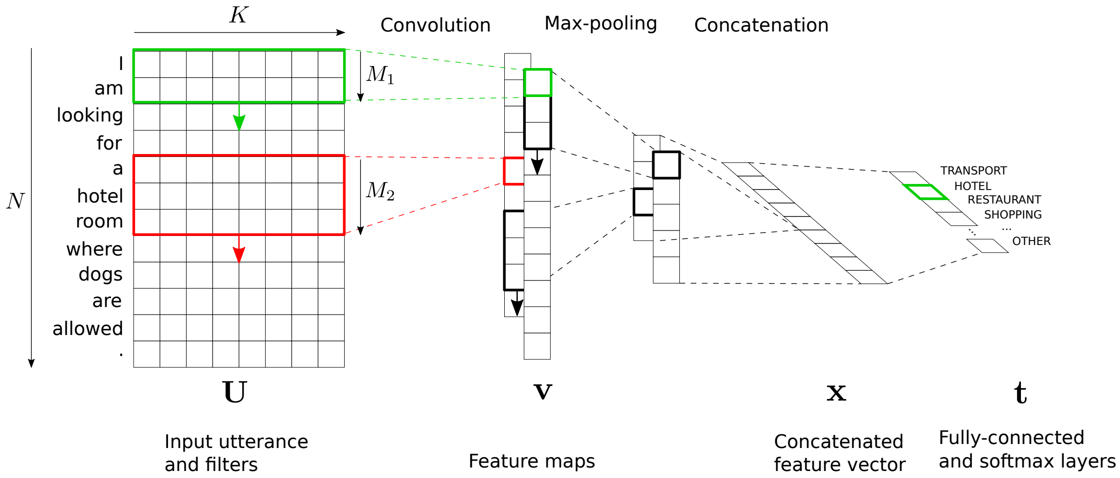


Figure 6.1: Illustration of the CNN architecture for classifying utterances into topic classes [Kim, 2014]. The input **U** is a $(N \times K)$-dimensional matrix representing the concatenated word embeddings of an utterance. The CNN is composed of two kernels with a max-pooling of size three.

However, the CNN alone does not take into consideration previous utterances when performing the classification, thus failing Aim 3. This limitation is addressed by the LRCN architecture used in Kim et al. [2016] discussed in the next section.

6.1.2 Topic Classification and Tracking from Dialogue History

Building on the CNN-based approach of Kim [2014] to classify single utterances into topic classes (Section 6.1.1), the Kim et al. [2016] model additionally accounts for the dialogue history up to that particular utterance. To do so, the model uses a recurrent architecture to capture the dependencies between the feature vector of each utterance which has been computed via the CNN. In particular, the authors investigated two increasingly more sophisticated recurrent architectures, namely, vanilla RNN [Elman, 1990] and LSTM [Hochreiter and Schmidhuber, 1997]. Both architectures take as input an utterance feature vector $\mathbf{x}_t$, and outputs a hidden state $\mathbf{h}_t$ at each time step $t \in [1, N]$. In particular, a vanilla RNN computes its hidden state $\mathbf{h}_t$ as follows

$$\begin{aligned}\mathbf{h}_t &= \mathrm{RNN}\left(\mathbf{x}_t, \mathbf{h}_{t-1}\right) \\ &= g\left(\Theta_{x,h}\mathbf{x}_t + \Theta_{h,h}\mathbf{h}_{t-1} + b_h\right)\end{aligned}$$

where g is an activation function and the weights Θ and bias b are shared across time-steps. While vanilla RNNs are theoretically capable of handling long-term dependencies, it has been shown that training them with gradient descent leads to sub-optimal solutions [Bengio et al., 1994]. In our setting, dialogues may exhibit long-range dependencies

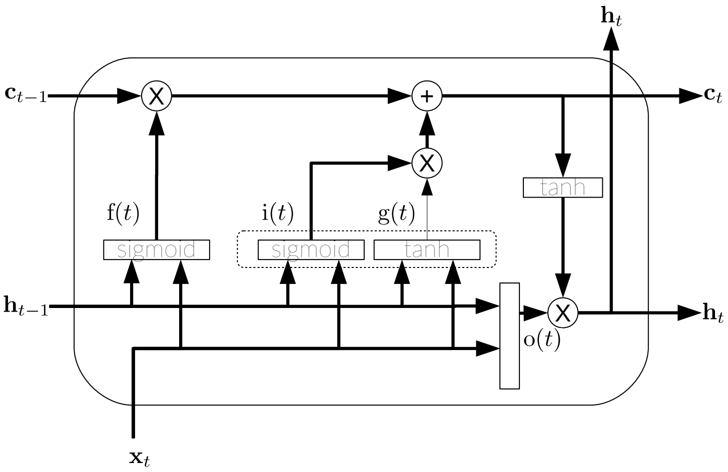


Figure 6.2: LSTM unit [Hochreiter and Schmidhuber, 1997].

between utterances that are not adjacent. Dialogue participants may refer back to an earlier sub-dialogue (e.g. when comparing alternatives, or discussing a complex subject) [Schulz et al., 2017]. LSTM [Hochreiter and Schmidhuber, 1997] solve this issue by making use of a number of *gates* to regulate the flow of information across timesteps (Figure 6.2).

In particular, an LSTM unit has a second output $\mathbf{c}_t$ called *cell state* vector in addition to the hidden state vector $\mathbf{h}_t$ such that

$$\mathbf{h}_t, \mathbf{c}_t = \text{LSTM}\left(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right)$$

where $\mathbf{c}_{t,i} \in [0, 1]$. The cell state vector $\mathbf{c}_t$ has the same number of dimensions as $\mathbf{h}_t$. In turn, the cell state $\mathbf{c}_t$ and the hidden state $\mathbf{h}_t$ are given by

$$\begin{cases} \mathbf{c}_t & = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \\ \mathbf{h}_t & = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right) \end{cases}$$

where the operator $\odot$ performs an element-wise multiplication. The three gates $\mathbf{f}_t$, $\mathbf{i}_t$ and $\mathbf{o}_t$ are real vectors protecting and controlling the cell state $\mathbf{c}_t$ and are of the same dimension as both $\mathbf{c}_t$ and $\mathbf{h}_t$. In particular, the *forget gate* $\mathbf{f}_t$ selects the information that should be removed from $\mathbf{c}_t$ based on $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$

$$\mathbf{f}_t = \sigma\left(\Theta_{x,f}\mathbf{x}_t + \Theta_{h,f}\mathbf{h}_{t-1} + \mathbf{b}_f\right).$$

The *input gate* $\mathbf{i}_t$ controls what new information is going to be stored in $\mathbf{c}_t$ by first, creating a vector of new candidate values $\mathbf{g}_t$ using a $\tanh(.)$ layer

$$\mathbf{i}_t = \sigma\left(\Theta_{x,i}\mathbf{x}_t + \Theta_{h,i}\mathbf{h}_{t-1} + \mathbf{b}_i\right),$$

$$\mathbf{g}_t = \tanh\left(\Theta_{x,g}\mathbf{x}_t + \Theta_{h,g}\mathbf{h}_{t-1} + \mathbf{b}_g\right).$$

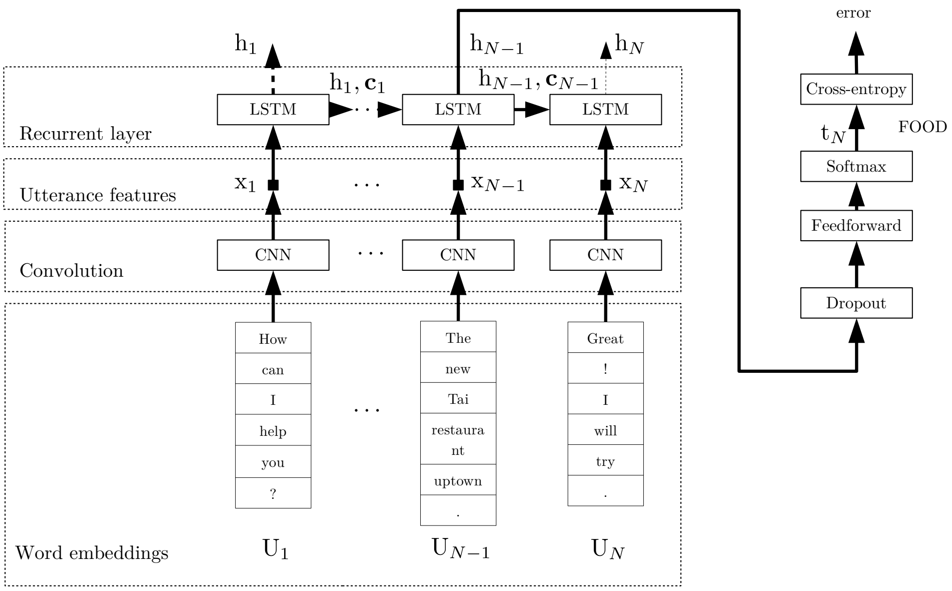


Figure 6.3: LRCN architecture combining a CNN and an LSTM for utterance classification [Kim et al., 2016]. During training, the model takes as input a set of utterances and outputs a sequence of topic classes.

And finally, the *output gate* $\mathbf{o}_t$ selects which part of $\mathbf{c}_t$ is going to the hidden state output $\mathbf{h}_t$

$$\mathbf{o}_t = \sigma\left(\Theta_{x,o}\mathbf{x}_t + \Theta_{h,o}\mathbf{h}_{t-1} + \mathbf{b}_o\right).$$

Similarly to the vanilla RNN, the weights Θ and biases $\mathbf{b}$ are shared across time-steps.

Now, to perform the topic classification, Kim et al. first connect the hidden state $\mathbf{h}_t$ of the LSTM to a dropout layer [Hinton et al., 2012] to improve on training performance and generalisation (Figure 6.3). Then a feedforward layer is added to compute the topic class scores, that are then fed into a softmax layer to produce the predictive distributions $q\left(t\right)$ over the topic classes. The output of the model at time t is the topic class $\mathbf{t}_t = \arg\max_t q\left(t\right)$ associated with the current utterance u_t. The model is trained end-to-end using stochastic gradient descent in a supervised learning fashion, using a cross-entropy loss function

$$H\left(p,q\right) = -\sum_t p\left(t\right)\log q\left(t\right) \tag{6.1}$$

between the ground truth topic class distributions $p\left(t\right)$ (i.e. a point-mass distribution) and the predicted topic class distributions $q\left(t\right)$ from the softmax layer. The benefit of training the model end-to-end is that the CNN learns the utterance features that are directly relevant to the topic classification.

Now, dialogues may cover a broad range of topics that do not always fall precisely into a pre-defined set of topic classes. We address this restriction in the following section where we introduce our contribution.

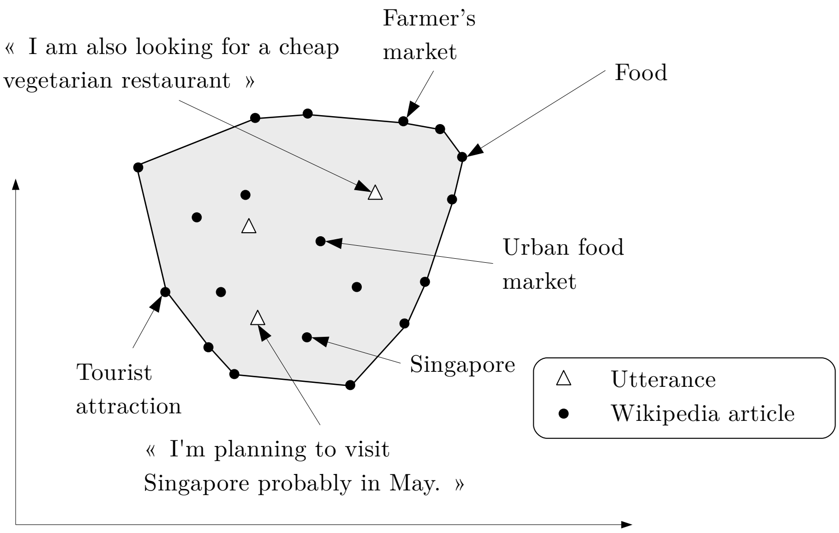

« I am also looking for a cheap vegetarian restaurant »
Farmer's market
Food
Urban food market
Tourist attraction
Singapore
« I'm planning to visit Singapore probably in May. »
△ Utterance
● Wikipedia article

Figure 6.4: Illustration of a two-dimensional semantic space with associated convex Hull of the topic embeddings (lines).

6.2 Joint Tracking of Topics and Topic Classes

Our proposed model extends the one presented in the previous section to deal with settings where utterances are assigned to both topic classes and topics. To do so, in addition to learning the topic class of each utterance, we also learn a continuous mapping of similarity between utterances and Wikipedia articles[2] under a common semantic space. Our method is based on the assumption that external encyclopedic knowledge from Wikipedia can be used to identify relevant topics for a given utterance. This assumption is common and has been employed in a number of prior related works [Vaart, 2000, Schonhofen, 2006, Banerjee et al., 2007, Breuing and Wachsmuth, 2012, Bhatia et al., 2016]. Because the encyclopedic knowledge in Wikipedia is constructed and maintained collaboratively by a large number of volunteers, it provides a huge amount of dynamically evolving information. Given this, our method consists of two main steps. In the first step, we automatically build a training set of utterance and Wikipedia article pairs. This is done offline prior to training our model. In the second step, we extend the structure of Kim et al. [2016] to interpolate the mapping identified in the previous step to new and unseen utterances, taking into account the dialogue history up to that utterance.

More specifically, in the first step, we compute a document embedding for each Wikipedia article using the doc2vec algorithm (Section 2.1.2). Since doc2vec is applicable to texts of any length (although longer semantic units yield more accurate vectors), it can readily be used to compute the embeddings of the Wikipedia articles in our setting (hereafter referred to as *topic embeddings*). These topic embeddings are computed once, separately from our model. Furthermore, independently of the topic embeddings' computation, we also rank the most relevant Wikipedia articles to each training utterance. To perform

[2] en.wikipedia.org

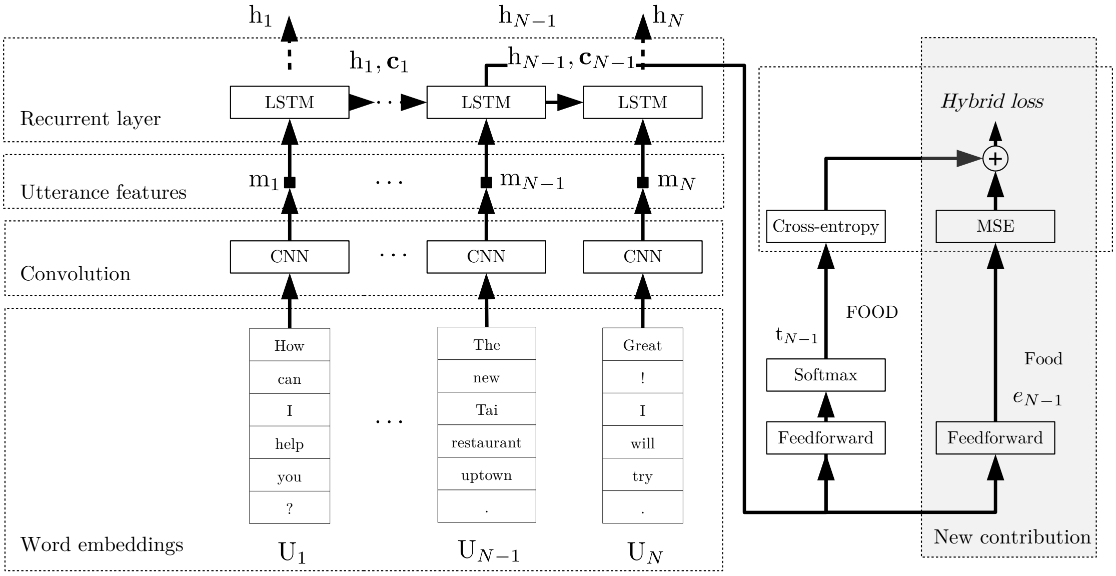


Figure 6.5: LRCN architecture combining a CNN and an RNN for utterance classification and regression. During training, the model takes as input a set of utterances and outputs a topic class and a topic embedding.

the ranking, we use the term frequency inverse document frequency (TF-IDF) algorithm [Manning et al., 2008]. TF-IDF is a prevailing technique in information retrieval and suited to our setting for its simplicity given the large amount of Wikipedia articles considered. Words in an utterance with a high TF-IDF score imply a strong relationship with the Wikipedia article they appear in. Now that we have both a topic embedding for each article and a ranking of Wikipedia articles per utterance, we can associate a training target for each utterance in the training set. Each training target for an utterance consists of the topic embedding of the top matching article or the average top-k topic embeddings.

In the second step, we extend the Kim et al. [2016] model to account for an unrestricted number of topics by performing a regression on the known target topic embeddings. This is achieved by adding a fully connected feedforward layer with linear activation function linked to the output of the LSTM (after the dropout layer) (Figure 6.5). Its role is to learn a mapping between the output of the LSTM and the target topic embeddings. In particular, the model learns to embed each utterance into the semantic space consisting of the topic embeddings constructed by the doc2vec algorithm (Figure 6.4). In such a semantic space, a continuous similarity measure (e.g. Euclidean distance or cosine similarity) is used to compute the distance between each utterance embedding and the closest topic embedding. When performing inference on unseen utterances, the solution lies within the convex Hull formed by topic embeddings in the training set (Figure 6.4). That is, the extent of the training data defines the solution space for the topic embeddings.

To train the model, we first concatenate all the dialogue sessions with each other, and then slide a context window across a fixed number of utterances. This prevents the model's complexity (i.e. its number of parameters) being dependent on the shape of the dataset, and in particular to the maximum length of the sessions which may be large. We assess the loss using a mean-squared error (MSE) objective function

$$\mathrm{MSE} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} (\mathbf{e}_i - \hat{\mathbf{e}}_i)^2 \tag{6.2}$$

between the predicted topic embedding **e** and the ground truth topic target $\hat{\mathbf{e}}$ for every utterance in the dataset $\mathcal{D}$. The model is jointly optimised end-to-end using a multi-objective learning function, encompassing errors not only from the topic classification (Equation 6.1), but also errors from topic regression (Equation 6.2).

6.3 Worked Example

To illustrate the process leading our topic tracker to jointly infer the topics and the topic classes[3], we turn to the example dialogue in Table 6.1. The dialogue is comprised of 15 utterances. We assume that the ground truth for the topic classes (Table 6.1, Column 4) has been identified by experts from a set of 5 possible alternatives (i.e. OPENING, HOTEL, FOOD, CLOSING, OTHER). We further consider a corpus of 10 Wikipedia articles to be used as alternatives for the topics (Table 6.2, Column 2). The ground truth for the topics (Table 6.1, Column 5) consists of the mapping of each utterance to a Wikipedia article. As such, we run the TF-IDF algorithm to produce this mapping based on the articles' content. In addition, our approach also requires the ground truth for the target topic embeddings. That is, the embedding associated with each Wikipedia article that will be used as target vector when computing the training loss for the regression. Therefore, we run the doc2vec algorithm on the Wikipedia corpus (with an abitrary vector size of 2) to generate the 10 embeddings (Table 6.2, Column 3). To aid visualisation, Figure 6.6 shows the resulting semantic space generated by these embeddings, along with its convex Hull region. After jointly training our model, we obtain a set of topic class predictions (Table 6.3, Column 2), and a set of topic embedding predictions (Table 6.3, Column 3). For the sake of simplicity in this short example, we are using the same 15 utterances to train the model and test its performance. We use the cosine similarity to identify the Wikipedia article from the predicted topic embedding

$$d_i = \arg\min_{d} \frac{e_i.\hat{e}_d}{|e_i|.|\hat{e}_d|}$$

for each predicted topic embedding e_i (Table 6.3, Column 3) and ground truth topic embedding $\hat{e}_i$ (Table 6.2, Column 3).

[3] The procedure is identical for the CNN (Section 6.1.1) and the LSTM (Section 6.1.2).

i	Speaker	Utterance (U_i)	Topic class	Topic
0	System	Hello, welcome to the San Francisco restaurant and hotel dialogue system. How may I help you?	OPENING	Greetings
1	User	I am looking for a hotel room where dogs are allowed	HOTEL	Pet–friendly hotel
2	System	Would you like a hotel that has internet?	HOTEL	Internet
3	User	I don't care	HOTEL	Internet
4	System	Where would you like it to be near to?	HOTEL	Location
5	User	A cheap hotel in the area of downtown	HOTEL	Location
6	System	Would you like a hotel that accepts credit cards?	HOTEL	Payment
7	User	I don't care	HOTEL	Payment
8	System	Grant hotel is a nice place, it is in the area of downtown, where no dogs are allowed , it is in the inexpensive price range, it has internet and it accepts credit cards.	HOTEL	Grant Hotel (San Francisco)
9	User	I am also looking for a cheap vegetarian restaurant	OTHER	Vegetarian cuisine
10	System	What part of town do you have in mind?	OTHER	Location
11	User	I don't care but it should be for lunch	OTHER	Location
12	System	Lucky creation restaurant is a nice place, it is good for lunch, it is in the cheap price range, it serves vegetarian food and it is in the area of chinatown.	OTHER	Vegetarian cuisine
13	User	Bye	CLOSING	Parting phrase
14	System	Goodbye.	CLOSING	Parting phrase

Table 6.1: Illustrative example of a dialogue session between a annotator and a dialogue system annotated with topics and topic classes (adapted from [Ultes et al., 2017]).

d	Title	Embedding ($\hat{e}_d$)
0	Greetings	[0.25, 0.20]
1	Pet-friendly hotel	[1.00, 0.30]
2	Internet	[0.15, 1.00]
3	Location	[0.35, 0.55]
4	Payment	[0.60, 0.65]
5	Grand hotel (San Francisco)	[0.90, 0.50]
6	Vegetarian cuisine	[0.85, 0.90]
7	Parting phrase	[0.10, 0.10]
8	Lunch	[1.05, 1.00]
9	Hotel	[0.95, 0.38]

Table 6.2: Pre-trained Wikipedia corpus displaying the learned doc2vec embeddings ($\hat{e}_d$).

i	Topic class	Topic	
	Prediction (t_i)	Prediction (e_i)	Closest Wiki. Article (d_i)
0	OPENING	[0.24, 0.18]	Greetings
1	HOTEL	[1.10, 0.20]	Pet-friendly hotel
2	HOTEL	[0.92, 0.40]	Hotel (*)
3	HOTEL	[0.84, 0.39]	Hotel (*)
4	HOTEL	[0.34, 0.50]	Location
5	HOTEL	[0.36, 0.53]	Location
6	HOTEL	[0.61, 0.66]	Payment
7	CLOSING (*)	[0.59, 0.64]	Payment
8	HOTEL	[0.91, 0.52]	Grant hotel (San Francisco)
9	FOOD (*)	[0.86, 0.89]	Vegetarian cuisine
10	FOOD (*)	[0.35, 0.52]	Location
11	FOOD (*)	[1.00, 0.95]	Lunch (*)
12	FOOD (*)	[0.87, 0.90]	Vegetarian cuisine
13	CLOSING	[0.11, 0.11]	Parting phrase
14	CLOSING	[0.09, 0.09]	Parting phrase

Table 6.3: Topic class and topic predictions after training on the illustrative example in Table 6.1. Predictions marked with a start (*) differ from the ground truth.

Given the predictions in Table 6.3, we now measure the performance of the model with respect to the ground truth. To do this, we use the multi-class classification metrics defined in Section 6.4.2. In more detail, we first compute the confusion matrix for the topic classes (Table 6.4a), and the topics (Table 6.4b) based on the predictions (Table 6.3, Column 2 and 4), and the ground truths (Table 6.1, Column 4 and 5).

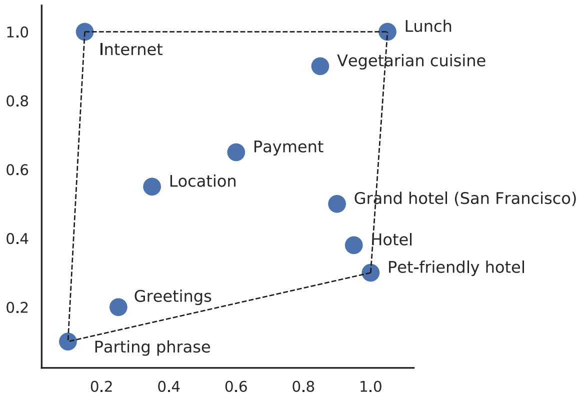


Figure 6.6: Semantic space and convex Hull of our example dialogue.

	OPEN	HOTEL	FOOD	CLOSING	OTHER
OPEN	**1**	0	0	0	0
HOTEL	0	**7**	0	**1**	0
FOOD	0	0	0	0	0
CLOSING	0	0	0	**2**	0
OTHER	0	0	**4**	0	0

(a) Topic classes.

	0	1	2	3	4	5	6	7	8	9
0	**1**	0	0	0	0	0	0	0	0	0
1	0	**1**	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	**2**
3	0	0	0	**3**	0	0	0	0	**1**	0
4	0	0	0	0	**2**	0	0	0	0	0
5	0	0	0	0	0	**1**	0	0	0	0
6	0	0	0	0	0	0	**2**	0	0	0
7	0	0	0	0	0	0	0	**2**	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

(b) Topics.

Table 6.4: Confusion Matrices.

From the confusions matrices in Table 6.4, we use Equation 6.3, Equation 6.4, Equation 6.5, and Equation 6.6 to compute the accuracy, precision, recall, and F1-score respectively (Table 6.5).

Topic Class				Topic			
A	P	R	F	A	P	R	F
0.67	0.53	0.57	0.55	0.8	0.7	0.67	0.68

Table 6.5: Results in terms of accuracy (A), precision (P), recall (R), and F1-score (F).

The accuracy results in Table 6.5 reflect the fewer overall number of prediction errors made on the topics (three) when compared to the topic classes (five). Furthermore, the average ratio of correct predictions among all the utterances with ground truth of a given topic (i.e. recall) is higher on the topics (0.67) than with topic classes (0.57). Similarly, the average ratio of correct predictions among all the utterances with a given predicted topic (i.e. precision) is higher on the topics (0.7) than with topic classes (0.53). These consistent results between the precision and recall are also reflected in the F1 score where the score for the topics (0.68) are higher than the topic classes (0.55).

6.4 Experimental Evaluation

In this section we report the performance results on both classification and regression in a real world setting. We first specify the datasets used in Section 6.4.1, and characterise the performance metrics in Section 6.4.2. We then describe the benchmarks in Section 6.4.3, and detail our experimental setting in Section 6.4.4. We finally discuss our results in Section 6.4.5.

6.4.1 Datasets

Our experiments use a total of two datasets.

- **TourSG.** This corpus (released as part of the DSTC4[4] competition) is composed of 35 manually transcribed dialogue sessions between tour guides and tourists in Singapore. Each of the 31,034 utterances has been annotated with one of nine domains: ATTRACTION (39.2%), TRANSPORTATION (13%), OTHER (12.7%), FOOD (12.4%), ACCOMMODATION (11.3%), SHOPPING (5.7%), ITINERARY (2.3%), CLOSING (1.7%) and OPENING (1.6%). The dataset has a vocabulary size of 6,035 words. The average length of an utterance is 9.25 ± 8.01 words, and the average length of a session is 887 ± 185 utterances.
- **Wikipedia.** The pre-processing of our training data requires the Wikipedia dataset[5]. The dataset is composed of 4.5 million articles of the English Wikipedia, and has a vocabulary size of 2 million words. No other information from the dataset was used other than the title and the raw text content of each article.

6.4.2 Performance Metrics

The performance of both topic classes identification and topics identification are assessed using standard multi-class classification measures. In particular, to identify the topics, we assess the models at recovering the Wikipedia articles identified by TF-IDF. That is, we first learn an embedding for all the utterances on the test set, then use the Euclidean distance to identify the nearest neighbouring topic embeddings from the Wikipedia articles. We label the learned embeddings as correctly classified if the nearest topic embedding matches with the one identified by TF-IDF, otherwise they are labelled

[4] http://www.colips.org/workshop/dstc4/

[5] https://en.wikipedia.org/wiki/Main_Page

as an incorrect classification. The classification metrics used in our evaluation take into account either the entire dataset (accuracy) or only the relevant subsets of the dataset (precision, recall, and F1-score). Given the confusion matrix in Equation 3.12 where i represents the ground-truth, j the prediction, and J the number of alternatives , the accuracy is given by

$$\text{Accuracy} = \frac{\sum_i \pi_{i,i}}{\sum_{i,j} \pi_{i,j}}, \tag{6.3}$$

the unweighted mean precision by

$$\begin{aligned} \text{Precision} &= \frac{1}{J} \sum_j \text{Precision}_j \\ &= \frac{1}{J} \sum_j \left(\frac{\pi_{j,j}}{\sum_i \pi_{i,j}} \right), \end{aligned} \tag{6.4}$$

the unweighted mean recall by

$$\begin{aligned} \text{Recall} &= \frac{1}{J} \sum_i \text{Recall}_i \\ &= \frac{1}{J} \sum_i \left(\frac{\pi_{i,i}}{\sum_j \pi_{i,j}} \right), \end{aligned} \tag{6.5}$$

and the unweighted mean F1-score by

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{6.6}$$

6.4.3 Benchmarks

We compare performance against each component of our proposed architecture. The strength of each benchmark lies in the potential inclusion of the dialogue history, and the mechanism by which the utterance features are calculated.

- **CNN.** This benchmark does not take into account the dialogue history. It first computes the utterance features, and then performs classification and/or regression using a feedforward layer followed by a softmax layer for the classification, and a feedforward layer alone for the regression.
- **LSTM.** This benchmark takes into account temporal dependencies between utterances. However, instead of taking as input the utterance features computed from the CNN, it uses pre-trained embeddings from the doc2vec algorithm.
- **Random.** This benchmark assigns to each utterance a topic class and a Wikipedia article at random with equal probabilities.

Each benchmark is trained in three different configurations: domain classification only (D), topic regression only (T), and both[6] (D+T).

6.4.4 Experimental Setting

To ensure generalisation (Aim 4) and avoid a dependence of our model's parameters to the structure of the training dataset, we concatenate the 35 dialogues sessions of the TourSG dataset into a single contiguous sequence of 31,034 utterances. This enables us to use a context window of fixed size irrespective of the number and length of each dialogue session. For this reason, our approach is applicable to any dialogue dataset, provided a topic and a topic class are assigned to each utterance. It is worth noting that by concatenating the TourSG dataset, the context window will intermittently overlap with opening and closing utterances from adjacent dialogue sessions at training time. Although this consequence represents a small fraction of the training data, it may impact prediction performance of the opening and closing utterances at test time. One approach would be to pad each dialogue sessions by a small but fixed amount (depending of the context window's size) of missing values (i.e. NULL) prior to concatenation. For example, a context window of 10 utterances would require a padding of 9 missing values between each dialogue session.

As such, we unroll the LSTM for 20 timesteps which is a reasonable range for recurrent architectures to perform well [Bahdanau et al., 2014]. We train the models in a supervised setting and divide our collection of 31,034 contiguous utterances into training (60%, i.e. 18,620 utterances), validation (20%, i.e. 6,207 utterances), and test sets (20%, i.e. 6,207 utterances). We further set the hidden state size of the LSTM to 300, the embedding size to 200, and the batch size to 5 utterances. We use 64 filters of height 1 and stride 1, with global max-pooling for the CNN. The CNN and LRCN are initialised with pre-trained GloVe[7] word embeddings (Section 2.2) for faster convergence. We use the implementation of the doc2vec algorithm provided by the Gensim[8] library to compute the pre-trained topic embeddings and utterance embeddings for the LSTM benchmark. Furthermore, we make use of PV-DM variant of the doc2vec algorithm as it has been shown to consistently perform better than PV-DBOW [Le and Mikolov, 2014]. We exclude Wikipedia articles of less than 50 words due to the limitations of doc2vec with short-length documents [De Boom et al., 2015]. We use the Gensim's implementation of the TF-IDF algorithm to map the utterances to the Wikipedia articles. After this mapping, we are left with 4,409 unique Wikipedia articles out of the 4.5 million initial candidates. We use the topic embedding of the top matching Wikipedia article ($k = 1$) as training target for each utterance. Finally, we set the dropout probability to 80%, and elect the Adam optimiser [Kingma and Ba, 2014] with a learning rate to 0.001.

[6] Excluding the random benchmark.

[7] https://nlp.stanford.edu/projects/glove/

[8] https://radimrehurek.com/gensim/

6.4.5 Results

Table 6.6 compares the performance of our model and the benchmarks when trained in the three different configurations: topic classification only, topic regression only, and both. We first observe that all models outperform the random benchmark by up to 20%. In particular, we observe that the challenge of predicting the correct Wikipedia article is so significant that the random benchmark is unable to recover any articles at all, achieving an accuracy, recall and precision of zero. Such a task is in effect equivalent to performing a multi-class classification with about four thousand alternatives (i.e. the number of unique Wikipedia articles in the training set) compared to the more manageable nine alternatives in the topic class prediction problem. Furthermore, we note that the combined LRCN approach clearly outperforms each of its component in isolation across all training configurations. This confirms that the CNN is indeed learning useful utterance features, that in turn sees their dependencies in time successfully captured by the LSTM. In addition, our approach consisting of training the LRCN jointly on classification and regression gives comparable results on the topic class prediction alone, but more importantly, the joint training significantly improves the performance of the more challenging task of identifying topics via regression by up to about 30%.

	Topic Classes				Topics			
Models	A	P	R	F	A	P	R	F
Random (D)	11.11	10.97	10.81	8.89	-	-	-	-
CNN (D)	50.17	56.48	50.17	47.61	-	-	-	-
RNN (D)	35.28	34.89	35.28	34.31	-	-	-	-
LRCN (D)	**51.44**	52.25	**51.44**	**50.78**	-	-	-	-
Random (T)	-	-	-	-	0	0	0	0
CNN (T)	-	-	-	-	17.82	15.77	17.82	16.33
RNN (T)	-	-	-	-	13.86	15.33	13.86	13.18
LRCN (T)	-	-	-	-	24.75	24.88	24.75	24.50
CNN (D+T)	51.09	**57.43**	51.09	48.26	18.81	16.04	18.81	17.19
RNN (D+T)	38.51	37.16	38.51	36.52	19.80	18.05	19.80	18.38
LRCN (D+T)	50.28	50.71	50.28	49.49	**30.69**	**31.55**	**30.69**	**30.24**

Table 6.6: Performance comparison between our approach (i.e. LRCN (D+T)) and the benchmarks. The letter(s) in parentheses indicate if training has been performed on domain classification only (D), topic regression only (T), or both (D+T). The letters A, P, R, F stands for accuracy, precision, recall, and F1-score respectively. The best performing models are highlighted in bold.

6.5 Summary

In this chapter, we introduced a novel architecture that for the first time simultaneously tracks both the topic class and the topic of each utterance in a dialogue session. Our key premises were that: (i) the handling of these two settings is essential to achieve both efficiency and accuracy in the response generated by downstream systems, and (ii) the title and content of Wikipedia articles are a good proxy for the topic of an utterance. Specifically, our approach builds a semantic space consisting of topic embeddings derived from Wikipedia articles within which new utterances are mapped. A subset of the Wikipedia articles used during training defines the bounds in the semantic space (i.e. the convex Hull) within which the predicted topics lies. This enables a very large set of candidate articles to be mapped to each utterance. We showed experimentally on a real-world dataset that our approach of jointly training the LRCN do not degrade domain prediction performance when compared to Kim et al. [2016], but more significantly, it is up to about 30% more accurate when predicting the nearest Wikipedia article.

We have identified several directions for future work that may be investigated to improve performance further. First, adding a measure of uncertainty in the topic predictions would lessen the impact of errors in downstream systems. In the absence of such assessments, downstream systems may take prediction errors as fact and act based on invalid information, thus incorrectly engaging in a wrong conversational strategy. Addressing this issue is challenging as the large number of parameters in our approach prevents the direct use of Bayesian inference techniques [MacKay, 1992, Pearce et al., 2018]. Second, a further enhancement over TF-IDF would be to use human expertise by combining crowdsourcing judgments to map the utterances to Wikipedia articles, thus improving on the labelling quality of the training dataset. In this setting, humans are more capable of assessing the semantics of both the utterances and the Wikipedia articles in the context of the dialogue. However, as discussed in Section 1.2 and 3.3, using crowdsourcing comes with its own set of unique challenges that would need to be addressed. Last, the training losses from the classification and regression operate on different scales. A further improvement would involve weighting the losses in a principled way rather than using heuristics. However, the competing nature of our training objectives would force us to model the trade-off between classification and regression, which is non trivial [Sener and Koltun, 2018].

Chapter 7

Summary and Future Direction

In this chapter, we present an summary of the research carried out in this thesis (Section 7.1), and suggests a new promising direction for future work (Section 7.2).

7.1 Summary

In this thesis, we looked at the field of topic identification at large through the lens of distributional semantics. In particular, we investigated three key problems, namely the evaluation of topic relevance, the learning of interpretable topics, and the identification and tracking of topics in written dialogues. Underlying our topic learning and identification solutions was the framework of distributional semantics which captures meaning directly from data as opposed to the handcrafted approach of formal semantics. Furthermore, we identified crowdsourcing as a practical and cost-effective technique to gauge the human interpretability and relevance of the learned topics.

More specifically, in Chapter 2, the relevant literature in the field of distributional semantics was reviewed. Emphasis was given to topic identification methods based on prediction and count models of distributional semantic (DSMs). We further identified a number of opportunities and challenges offered by DSMs. First, we observed that topic models may miss or infer incorrect associations between topics and documents. In light of this, we identified crowdsourcing as a practical and cost-effective technique to perform human evaluation of topic relevance. Nonetheless, despite the potential advantages of crowdsourcing, we saw that additional challenges related the quality of annotations, and in particular spam responses, needed to be addressed. Second, we acknowledged the difficulties that topic models face when generating topics that need to be human-interpretable. We therefore identified sentences as a richer form of semantic units to express topics than single words. Third, we pinpointed a number of challenges when tracking topics over time in dialogue systems, namely, the restrictive aspect of classifying utterances into small sets of topics known a priori, and the careful consideration of context.

In Chapter 3, we identified a number of approaches to combine unreliable judgments from annotators to evaluate topic relevance. As such, we selected IBCC as a promising point of departure because it models the annotators' bias in addition to their accuracy. We presented IBCC in the context of aggregating judgments regarding documents with a single topic. We then experimentally evaluated IBCC when aggregating judgments from documents with multiple topics. Results on synthetic data showed that when faced with judgments from documents with multiple topics, IBCC provides inaccurate aggregation and inference of the annotators' accuracy and bias.

In Chapter 4, we addressed the shortcomming of IBCC and extended its architecture to reliably aggregate judgments of topic relevance for document with multiple topics from unreliable annotators. Motivated by the ability of confusion matrices at capturing detailed performance assessments of the participants, our key insight was to sample both the participants judgments and the topic proportions. Experimental results on three real-world datasets showed major improvements on three key aspects compared to existing methods. First, we achieved up to 28% improvement in terms of aggregation accuracy. Second, we showed robustness against spammers with comparable level of accuracy when 60% of the participants are spammers, as other approaches do when there are no spammers. Third, we demonstrated a five-fold decrease in the expected number of misclassified spammers.

In Chapter 5, we addressed another shortcoming of topic models, namely topic intepretability. One of the key innovations of our method was the leveraging of context to learn more interpretable topics than unigram-based topic models. In particular, sentences were used as a richer form of semantic units to express topics. Furthermore, another key aspect of our approach was that, when presented with new documents, no re-training was required to infer the topics. We showed that our relatively unoptimised approach performed well in text classification tasks with up to 70% accuracy. Moreover, we introduced a novel evaluation study measuring the ability of human participants at establishing thematic relations between books and topics when presented as lists of sentences rather than lists of words. The design of our human evaluation framework captured a practical usage of topic models and is a step forward in validating document modeling assumptions.

Finally, in Chapter 6, we introduced a novel topic tracking architecture that for the first time simultaneously classify the topic of each utterance in a dialogue session while identifying the title of the most relevant Wikipedia article from a large set of candidates. The handling of the dialogue history was crucial to our approach since it provided context to short utterances (e.g. "ok.", "thank you!", or "yes.") that may otherwise be difficult to process. Experimental results on real human-to-human dialogues showed that our approach generates comparable performance when identifying the topic classes than the components of the architecture in isolation. But more significantly, our joint training achieves more accurate predictions of the nearest Wikipedia article by up to about 30%.

7.2 Future Direction

We observed that the unsupervised nature of DSMs provides no guarantees that the topics discovered correlate with human judgment. Furthermore, we also observed that DSMs experience degradation in accuracy over texts of short length, or when rare words are used (i.e. jargon or slang) due to the lack of word co-occurrence information. Such loss in accuracy makes it challenging to learn reliable distributed representations and lead to imprecise results when used as feature to classify topics. Despite recent advances in DSMs, most effective approaches today have focused on a single workflow where human involvement is often seen as external to the architecture. Therefore, a promising long-term research direction is to adopt a human-in-the-loop approach [Munro, 2020], where both humans and algorithms are leveraged in a virtuous circle of training, tuning, and testing to improve on the accuracy of topic identification. Such approach combines the best of human intelligence (e.g. common sense or background knowledge) with the best of machine intelligence (e.g. patterns identification in large datasets or processing speed).

In practice, beyond the eventual initial labelling of training data required for supervised learning, human-in-the-loop approaches relies heavily on active learning [Settles, 2010] where data is continuously tuned by humans and fed back into the algorithm in order to achieve better results. In such a workflow, the model produces suggestions of topics to be presented to humans for evaluation while being trained. The suggested topics may include topics where the model is unconfident, or where the model is over confident about incorrect topics. Humans review the results and correct any inaccuracies that the model may produce. This feedback is then presented back to the model. The goal is to fine-tune the model in order to make better future decisions, and eventually reach correct results without human intervention.

It is important to note that to achieve the broad impact that distributional semantics has the potential to make, it is critical to democratise human involvement as trained experts are less likely to be available. However, incorporating human knowledge and feedback from non-experts into machine learning algorithms still remains a major open research questions. Nonetheless, we believe that such human-in-the-loop interactions will define most uses of machine learning in the years to come.

Appendix A

The Dirichlet Distribution

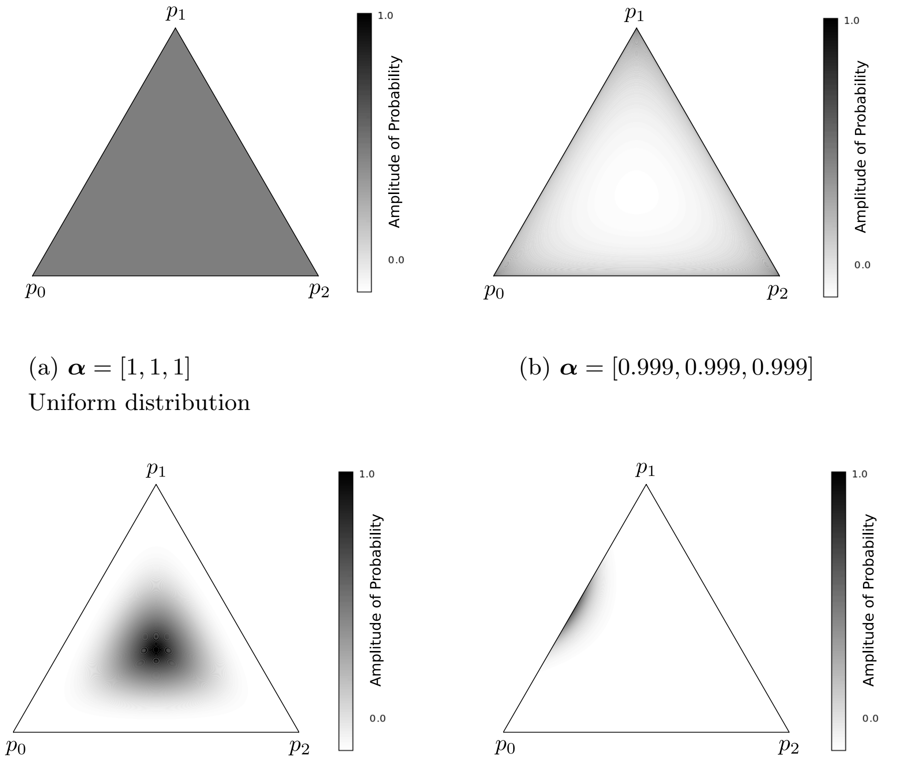
In this appendix, we present the Dirichlet distribution which is extensively used in this thesis. A Dirichlet distribution is a continuous distribution over vectors $\mathbf{p} = [p_0, \cdots, p_K]$ of $K \geq 2$ real numbers, each summing to 1,

$$\mathbf{p} \sim \text{Dirichlet}(\boldsymbol{\alpha}).$$

The *concentration* parameter is the vector $\boldsymbol{\alpha} = [\alpha_0, \cdots, \alpha_K]$ of dimensions K which determines how "concentrated" the samples from the Dirichlet distribution are likely to be. The density function of Dirichlet distribution is given by

$$f\left(\mathbf{p}; \boldsymbol{\alpha}\right) = \frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}\right)} \prod_{i=1}^{K} p_i^{\alpha_i - 1}, \tag{A.1}$$

where $\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}$ is a normalising constant, $\mathrm{B}\left(\boldsymbol{\alpha}\right) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}$ the multinomial multivariate Beta function, $\Gamma\left(z\right) = \int_0^\infty t^{z-1} e^{-t} dt$ the gamma function, and $\sum_{i=1}^{K} y_i = 1$ and $y_i \geq 0$ for all $i \in [1, K]$. Figure A.1 shows a Dirichlet distribution for different values of $\boldsymbol{\alpha}$ displayed for $K = 3$. Each point within a triangle represent an assignment of $\boldsymbol{p}$ sampled from Equation A.1.



(a) $\boldsymbol{\alpha} = [1, 1, 1]$
Uniform distribution

(b) $\boldsymbol{\alpha} = [0.999, 0.999, 0.999]$

(c) $\boldsymbol{\alpha} = [5, 5, 5]$

(d) $\boldsymbol{\alpha} = [2, 5, 15]$

Figure A.1: Three-dimensional Dirichlet distribution for several different values of $\boldsymbol{\alpha}$ displayed on three-dimensional *simplexes* (i.e. the generalisation of a triangle to multiple dimensions). The corners of the simplexes correspond to three distributions where each label has probability one. Each point of the simplexes correspond to a different distributions.

Expectation	Mode	Variance	Covariance
$E[p_i] = \frac{\alpha_i}{\hat{\alpha}}$	$\text{mode}(p_i) = \frac{\alpha_i - 1}{\hat{\alpha} - K}$	$\text{var}(p_i) = \frac{\alpha_i(\hat{\alpha} - \alpha_i)}{\alpha_0^2(\hat{\alpha}+1)}$	$\text{cov}(p_i, p_j) = -\frac{\alpha_i \alpha_j}{\alpha_0^2(\alpha_0+1)} \ (i \neq j)$

Table A.1: First two moments and mode of the Dirichlet distribution where $\hat{\alpha} = \sum_{i=1}^{K} \alpha_i$.

Dirichlet distributions are very often used as prior distributions of the multinomial distribution in Bayesian inference. In this contex, $\mathbf{p}$ is interpreted as the parameter of a multinomial distribution and $\boldsymbol{\alpha}$ as the *pseudo-counts* of prior observations of the K categories.

A.1 Logarithmic Form

The logarithm of a Dirichlet distribution is given by

$$\begin{aligned} \ln p\left(\mathbf{p};\boldsymbol{\alpha}\right) &= \ln\left(\frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}\right)}\prod_{i=1}^{K} p_i^{\alpha_i-1}\right) \\ &= \sum_{i=1}^{K}\left(\alpha_i-1\right)\ln p_i - \ln\mathrm{B}\left(\boldsymbol{\alpha}\right). \end{aligned} \tag{A.2}$$

Furthermore, the expectation of the logarithm of the i-th component of random variable $\mathbf{p}$ is given by

$$E\left[\ln p_i\right] = \psi\left(\alpha_i\right) - \psi\left(\hat{\alpha}\right), \tag{A.3}$$

where $\psi\left(x\right) = \frac{d}{dx}\ln\left(\Gamma\left(x\right)\right) = \frac{\Gamma'(x)}{\Gamma(x)}$ is the digamma function.

A.2 Conjugate Prior

Proposition A.1. (Dirichlet Conjugate Prior) *Given a model*

$$\mathbf{p} \sim \textit{Dirichlet}\left(\boldsymbol{\alpha}\right)$$

$$\mathbf{x}|\mathbf{p} \sim \textit{Multinomial}\left(n,\boldsymbol{p}\right)$$

with random variables $\mathbf{x} = [x_1,\cdots,x_m]$*, a probability vector* $\mathbf{p} = [p_1,\cdots,p_m]$*, and hyperparameter* $\boldsymbol{\alpha} = [\alpha_1,\cdots,\alpha_m]$*, the posterior distribution of* $\mathbf{p}$ *given* $\mathbf{x}$ *and* $\boldsymbol{\alpha}$ *is again Dirichlet distributed with parameter* $\boldsymbol{\alpha}+\mathbf{x}$

$$\mathbf{p}|\mathbf{x};\boldsymbol{\alpha} \sim \textit{Dirichlet}\left(\boldsymbol{\alpha}+\mathbf{x}\right).$$

In other words, the Dirichlet distribution is a conjugate prior for the multinomial distribution.

Proof. The posterior distribution is given by the Bayes rule

$$p\left(\mathbf{p}|\mathbf{x};\boldsymbol{\alpha}\right) = \frac{1}{p\left(\mathbf{x};\boldsymbol{\alpha}\right)}p\left(\mathbf{x}|\mathbf{p}\right)p\left(\mathbf{p};\boldsymbol{\alpha}\right)$$

where $p\left(\mathbf{x};\boldsymbol{\alpha}\right)$ is the *Dirichlet-multinomial distribution.*

Lemma A.2. *The* Dirichlet-multinomial distribution *is given by*

$$p\left(\mathbf{x};\boldsymbol{\alpha}\right) = \begin{pmatrix} n \\ \mathbf{x} \end{pmatrix}\frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}\right)}\mathrm{B}\left(\mathbf{x}+\boldsymbol{\alpha}\right)$$

where $\begin{pmatrix} n \\ \mathbf{x} \end{pmatrix}$ *is the multinomial coefficient* and $\mathrm{B}(.)$ *the multivariate Beta function acting as normalising constant.*

Proof. From the Bayes rule and the definition of $\mathbf{x}$ and $\mathbf{p}$

$$\begin{aligned}
p\left(\mathbf{x};\boldsymbol{\alpha}\right) &= \int p\left(\mathbf{x}|\mathbf{p}\right)p\left(\mathbf{p};\boldsymbol{\alpha}\right)d\mathbf{p} \\
&= \int \left(\begin{pmatrix} n \\ \mathbf{x} \end{pmatrix}\prod_{i=1}^{m} p_i^{x_i}\right)\left(\frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}\right)}\prod_{i=1}^{m} p_i^{\alpha_i-1}\right)d\mathbf{p} \\
&= \begin{pmatrix} n \\ \mathbf{x} \end{pmatrix}\frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}\right)}\int\prod_{i=1}^{m} p_i^{x_i+\alpha_i-1}d\mathbf{p}.
\end{aligned}$$

To solve the integral, we transform the integrand into a Dirichlet distribution (which integrate to 1), such that

$$\begin{aligned}
\int\prod_{i=1}^{m} p_i^{x_i+\alpha_i-1}d\mathbf{p} &= \mathrm{B}\left(\mathbf{x}+\boldsymbol{\alpha}\right)\int\frac{1}{\mathrm{B}\left(\mathbf{x}+\boldsymbol{\alpha}\right)}\prod_{i=1}^{m} p_i^{x_i+\alpha_i-1}d\mathbf{p} \\
&= \mathrm{B}\left(\mathbf{x}+\boldsymbol{\alpha}\right)
\end{aligned}$$

□

Combining the terms gives

$$\begin{aligned}
p\left(\mathbf{p}|\mathbf{x},\boldsymbol{\alpha}\right) &= \frac{1}{\cancel{\begin{pmatrix} n \\ \mathbf{x} \end{pmatrix}}\cancel{\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}}\mathrm{B}\left(\mathbf{x}+\boldsymbol{\alpha}\right)}\left(\cancel{\begin{pmatrix} n \\ \mathbf{x} \end{pmatrix}}\prod_{i=1}^{m} p_i^{x_i}\right)\left(\cancel{\frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}\right)}}\prod_{i=1}^{m} p_i^{\alpha_i-1}\right) \\
&= \frac{1}{\mathrm{B}\left(\mathbf{x}+\boldsymbol{\alpha}\right)}\prod_{i=1}^{m} p_i^{x_i+\alpha_i-1}.
\end{aligned}$$

□

Appendix B

Exerpts from the Snow et al. Dataset

This appendix presents a sample of 128 tasks with corresponding jugdments and ground truth from the sentiment analysis dataset used in Chapter 4. The full dataset is available publicly at https://sites.google.com/site/nlpannotations/.

- **task_id.** task identifier
- **worker_id.** worker identifier
- **anger_judg, disgust_judg, fear_judg, joy_judg, sadness_judg, surprise_judg, valence_judg.** workers' jugments
- **anger_gold, disgust_gold, fear_gold, joy_gold, sadness_gold, surprise_gold, valence_gold.** ground truth

task_id	text	anger_gold	disgust_gold	fear_gold	joy_gold	sadness_gold	surprise_gold	valence_gold
500	Test to predict breast cancer relapse is approved	0	0	15	38	9	11	32
501	Two Hussein allies are hanged, Iraqi official says	24	26	16	13	38	5	-48
502	Sights and sounds from CES	0	0	0	17	0	4	26
503	Schuey sees Ferrari unveil new car	0	0	0	46	0	31	40
504	Closings and cancellations top advice on flu outbreak	1	0	23	8	11	8	-6
505	Trucks swallowed in subway collapse	8	0	28	0	57	7	-67
506	Sarkozy letter surprises French cartoons hearing	0	0	0	0	0	53	14
507	Building a memorial to a son, one child at a time	10	0	9	22	36	3	-9
508	Lawmaker seeks iPod ban in crosswalks in New York	11	0	0	0	4	38	-15
509	Diabetic waits months for eyeglasses	34	11	0	0	40	13	-32
510	Sudan tells United Nations envoy to leave in 72 hours	18	0	0	0	15	59	-34
511	5000 years on but couple still hugging	0	0	0	53	0	65	61
512	Defense to challenge Russert's credibility	4	5	0	0	0	19	-6
513	Ozzy, a Hero for the hard-rocking masses	0	0	0	37	0	4	34
514	CIA leak trial summary	9	17	2	0	4	33	-23
515	Dance movie takes over No. 1	0	0	0	56	0	37	63
516	Asian nations urge Myanmar reform	0	0	0	17	0	19	14
517	After Iraq trip, Clinton proposes war limits	8	0	8	53	13	25	38
518	7 dead in apartment building fire	14	2	47	0	86	10	-86
519	Male sweat boosts women's hormone levels	0	14	0	6	0	75	18
520	Carphone Warehouse's mixed signals	2	2	0	0	0	10	-2
521	Democrats plot Bush troop increase censure	19	1	1	19	9	21	-2
522	Cisco sues Apple over iPhone name	51	7	0	0	3	27	-46
523	BB star Jackson denies Goody comments	7	8	0	0	0	2	-10
524	Inter Milan set Serie A win record	2	0	0	50	0	9	50
525	US Airways boosts bid for Delta	0	10	0	18	2	26	23
526	Press sees hope in Mecca talks	0	0	0	60	3	16	55
527	Bears fan loses bet and changes name	2	13	0	2	17	33	-13
528	Global National Major child porn ring bust	46	64	0	42	31	27	13
529	'Human hair' clue in hunt for airliner	0	0	0	2	0	36	27
530	Johnny Depp to make movie of spy poisoning	0	0	0	40	8	17	36
531	Really?: The claim: the pill can make you put on weight	0	8	0	5	0	49	2
532	5 money makeovers	0	0	0	31	0	8	38
533	TBS to pay $2M fine for ad campaign bomb scare	25	32	45	11	28	43	-29
534	Stomp the Yard has winning moves in its weekend debut	0	0	0	68	0	13	62
535	Discovered boys bring shock, joy	0	0	3	55	0	56	45
536	Bernhard set to leave Volkswagen	11	0	0	0	40	22	-46
537	Pacers' Jackson misses gun hearing	14	10	24	2	3	21	-44
538	Cases: when the simple solution is the right one	0	0	0	15	0	12	27
539	Two Muslim groups sue French newspaper International	40	8	0	7	26	36	-32
540	Golden Globes on their way	0	0	0	63	0	0	60
541	At New OZZFEST, Freedom Ain't Free	11	0	5	6	1	18	-18
542	Federer handed tough Aussie draw	0	0	0	4	8	4	4
543	Protesters end strike as Nepal PM concedes demands	24	0	3	48	6	18	29

Continued on next page

task_id	text	anger_gold	disgust_gold	fear_gold	joy_gold	sadness_gold	surprise_gold	valence_gold
544	Turner pays for Boston "bombing"	17	0	30	2	0	16	-33
545	Two detained in body parts mailing	4	61	30	8	14	5	-30
546	Essay: about that mean streak of yours: psychiatry can do only so much	8	0	6	10	20	17	3
547	Hussein's niece pleads for father's life	18	19	17	0	53	12	-39
548	Sarkozy heads for clash in cartoons row	30	13	8	0	2	9	-44
549	Global web of suspects in child-rape download	55	54	0	0	36	13	-64
550	Blake defeats Moya to retain title	4	0	0	33	1	2	29
551	Storms kill, knock out power, cancel flights	3	9	82	0	60	0	-83
552	Aquarium puts ailing beluga whale to sleep	1	0	0	7	67	12	-44
553	Microsoft, Sony, we have a problem	0	0	6	0	21	26	-28
554	A police state? The issues	15	5	46	0	8	0	-43
555	Move to ban iPods from crossing the street	17	0	0	13	11	20	-10
556	India's Taj Mahal gets facelift	0	0	0	65	0	23	67
557	Cheney to Congress: Can't run Iraq war by committee	47	4	26	0	21	6	-56
558	Ganguly handed India squad call-up	4	4	6	31	5	12	6
559	Fortune: Looking beyond the iPhone	0	0	0	36	0	24	42
560	Really?: The claim: the back seat of a car is the safest place to sit	0	0	6	23	0	44	31
561	Vaccine mandate upsets legislators	17	8	11	13	7	5	-34
562	News analysis: Iranian boast is put to test	17	7	28	2	1	2	-24
563	Panel issues bleak report on climate change	20	15	48	0	45	0	-42
564	Virtual 'American Idol' hits right notes	0	0	0	44	0	25	51
565	Iraq car bombings kill 22 People, wound more than 60	32	27	84	0	95	20	-98
566	'Stomp' steps to No. 1 at box office	0	0	0	60	0	19	62
567	Google executive acts as goodwill envoy	0	0	0	50	0	29	50
568	Filipino woman kidnapped in Nigeria	32	15	77	0	50	9	-82
569	Seahawks, Bears vie for shot at NFL title	0	0	8	32	0	5	31
570	Brown not buried as estate issues loom	6	17	0	0	41	23	-43
571	Will Rob Cohen Direct Third 'Mummy'?	0	0	0	32	0	35	32
572	EU will urge China to go green	8	0	5	34	3	12	26
573	17th-Century Remedy; 21st-Century Potency	0	0	0	0	0	14	16
574	Forecasters expect toasty 2007	13	0	33	20	23	16	-18
575	'Sunshine' Goydos wins Sony open	0	0	0	75	0	10	81
576	Teacher charged with sex assault	52	87	18	0	23	25	-88
577	Merck: Gardasil may fight more strains	9	0	19	21	14	16	-2
578	Mountain glaciers melting faster, United Nations says	15	10	49	0	41	12	-52
579	Snow causes airport closures in Britain	6	0	17	0	22	13	-45
580	Bananaconda inventor is top poet of kids	0	0	0	63	0	24	62
581	Shareholders sue Apple	49	0	0	0	14	15	-52
582	Israelis retaliate after attack by Lebanese Army	48	13	53	0	40	0	-69
583	Deadly bird flu confirmed in British Turkeys	2	1	70	0	36	24	-72
584	Too little sleep may mean too fat kids	0	12	7	0	21	57	-51
585	Bathing mom awakes to find baby dead	2	0	32	0	93	46	-90
586	Visitors to Eiffel Tower climb to record in 2006	0	0	0	43	0	40	54
587	More sleep, healthier slimmer children	0	0	0	47	0	56	61
588	Queen battles Bond for Baftas	2	0	11	30	2	16	21

Continued on next page

task_id	text	anger_gold	disgust_gold	fear_gold	joy_gold	sadness_gold	surprise_gold	valence_gold
589	Walters on Trump: 'Poor, pathetic man'	23	25	0	0	28	12	-44
590	Ice storm smacks roads, power	7	10	75	0	29	2	-67
591	Iran says it will strike US interests if Attacked	52	13	67	0	22	22	-81
592	Palestinian factions to resume talks	0	0	0	41	2	26	43
593	Study: men's sweat triggers high sexual arousal in women	0	2	0	8	0	73	35
594	Outcry at N Korea 'nuclear test'	37	2	56	0	11	6	-66
595	Russia plans major military build-up	27	15	75	17	22	22	-45
596	Archaeologists find remains of couple locked in a hug	0	0	0	43	16	79	52
597	Stenson defends his title at Dubai	0	0	0	45	0	8	44
598	Taking the plunge	0	0	9	0	3	2	-10
599	Updates on world's top stories	0	0	0	14	0	2	12

worker_id	task_id	anger_judg	disgust_judg	fear_judg	joy_judg	sadness_judg	surprise_judg	valence_judg
A1AVJRFM6L0RN8	594	25	50	25	25	25	0	0
ADAGUJNWMEPT6	594	80	50	60	0	25	5	-60
A1LY3NJTYW9TFF	594	5	0	0	0	0	0	0
A14WWG6NKBDWGP	594	30	30	30	0	20	40	-50
A1VYRD3HO2WDUN	594	20	50	50	0	50	0	-50
A1XUURRBT9RYFW	594	10	10	50	0	20	10	-40
A1M0SEWUJYX9K0	594	60	60	60	0	75	20	-75
A2KBTYNGUFRB9N	594	80	80	60	0	0	30	-75
A3POYFULMTNW1H	594	25	0	20	0	10	0	-20
ARQ4J4TLTPBNC	594	100	100	40	0	100	0	-100
A1AVJRFM6L0RN8	582	0	0	0	0	0	0	0
ADAGUJNWMEPT6	582	5	5	5	0	45	10	-15
A1LY3NJTYW9TFF	582	0	0	0	0	0	0	0
A14WWG6NKBDWGP	582	50	40	40	0	80	70	-80
A1VYRD3HO2WDUN	582	0	0	40	0	40	0	-40
A1XUURRBT9RYFW	582	10	5	5	0	5	5	-10
A1M0SEWUJYX9K0	582	75	75	75	0	75	10	-75
A2KBTYNGUFRB9N	582	50	40	50	0	30	30	-60
A3POYFULMTNW1H	582	0	0	0	0	80	0	80
ARQ4J4TLTPBNC	582	100	100	0	0	100	0	-100
A1AVJRFM6L0RN8	593	0	0	0	75	0	75	75
ADAGUJNWMEPT6	593	0	15	0	10	0	15	15
A1LY3NJTYW9TFF	593	0	50	0	0	0	5	0
A14WWG6NKBDWGP	593	0	20	0	10	0	10	30
A1VYRD3HO2WDUN	593	0	50	0	0	0	80	-20
A1XUURRBT9RYFW	593	0	15	0	0	0	10	-5
A1M0SEWUJYX9K0	593	0	10	0	0	0	75	0
A2KBTYNGUFRB9N	593	0	30	0	10	0	90	20
A3POYFULMTNW1H	593	0	0	0	80	0	80	80
ARQ4J4TLTPBNC	593	0	0	0	0	0	0	0
A1AVJRFM6L0RN8	592	0	0	0	0	0	0	0
ADAGUJNWMEPT6	592	0	0	5	35	0	10	30

Continued on next page

worker_id	task_id	anger_judg	disgust_judg	fear_judg	joy_judg	sadness_judg	surprise_judg	valence_judg
A1LY3NJTYW9TFF	592	10	0	0	0	0	0	0
A14WWG6NKBDWGP	592	0	0	0	0	0	0	0
A1VYRD3HO2WDUN	592	0	0	0	10	0	5	5
A1XUURRBT9RYFW	592	0	0	5	5	0	5	5
A1M0SEWUJYX9K0	592	0	0	0	0	45	20	0
A2KBTYNGUFRB9N	592	0	0	30	60	0	100	50
A3POYFULMTNW1H	592	0	0	0	50	0	10	50
ARQ4J4TLTPBNC	592	0	0	0	100	0	0	100
A1AVJRFM6L0RN8	581	0	0	0	0	0	0	0
ADAGUJNWMEPT6	581	0	0	0	0	0	0	0
A1LY3NJTYW9TFF	581	40	40	0	0	0	0	0
A14WWG6NKBDWGP	581	50	50	0	0	0	60	-50
A1VYRD3HO2WDUN	581	0	10	0	0	20	0	-10
A1XUURRBT9RYFW	581	0	0	0	0	0	5	0
A1M0SEWUJYX9K0	581	25	50	0	0	50	10	-50
A2KBTYNGUFRB9N	581	20	30	0	0	0	40	-10
A3POYFULMTNW1H	581	0	0	0	30	0	30	30
ARQ4J4TLTPBNC	581	0	0	0	0	0	0	0
A1AVJRFM6L0RN8	591	75	75	25	0	0	0	0
ADAGUJNWMEPT6	591	50	45	25	0	5	5	-25
A1LY3NJTYW9TFF	591	20	5	0	0	0	0	0
A14WWG6NKBDWGP	591	40	0	40	0	60	50	-60
A1VYRD3HO2WDUN	591	0	0	80	0	40	0	-30
A1XUURRBT9RYFW	591	5	5	30	0	20	5	-20
A1M0SEWUJYX9K0	591	80	80	80	0	80	0	-100
A2KBTYNGUFRB9N	591	90	40	75	0	0	50	-90
A3POYFULMTNW1H	591	95	80	50	0	30	0	-80
ARQ4J4TLTPBNC	591	100	100	40	0	100	0	-100
A1AVJRFM6L0RN8	580	0	0	0	0	0	0	0
ADAGUJNWMEPT6	580	0	0	0	0	0	0	0
A1LY3NJTYW9TFF	580	0	0	0	5	0	0	0
A14WWG6NKBDWGP	580	0	0	0	80	0	50	75

Continued on next page

worker_id	task_id	anger_judg	disgust_judg	fear_judg	joy_judg	sadness_judg	surprise_judg	valence_judg
A1VYRD3HO2WDUN	580	0	0	0	5	0	0	5
A1XUURRBT9RYFW	580	0	0	0	5	0	0	5
A1M0SEWUJYX9K0	580	0	0	0	50	0	20	85
A2KBTYNGUFRB9N	580	0	0	0	80	0	70	80
A3POYFULMTNW1H	580	0	0	0	0	0	50	0
ARQ4J4TLTPBNC	580	0	0	0	100	0	10	0
A1AVJRFM6L0RN8	590	0	0	0	0	25	0	0
ADAGUJNWMEPT6	590	5	0	15	0	15	10	-15
A1LY3NJTYW9TFF	590	0	0	10	0	25	0	0
A14WWG6NKBDWGP	590	20	0	60	0	60	60	-60
A1VYRD3HO2WDUN	590	0	0	20	0	20	0	-20
A1XUURRBT9RYFW	590	0	0	0	0	5	0	-5
A1M0SEWUJYX9K0	590	0	0	50	0	30	0	-40
A2KBTYNGUFRB9N	590	0	0	60	0	80	60	-30
A3POYFULMTNW1H	590	0	0	30	0	0	0	-20
ARQ4J4TLTPBNC	590	0	0	0	0	0	0	0
A1AVJRFM6L0RN8	589	0	0	0	0	0	0	0
ADAGUJNWMEPT6	589	0	0	0	0	0	0	0
A1LY3NJTYW9TFF	589	30	10	0	0	0	30	-40
A14WWG6NKBDWGP	589	10	30	0	10	0	0	-10
A1VYRD3HO2WDUN	589	0	50	0	0	50	0	-20
A1XUURRBT9RYFW	589	0	0	0	5	0	10	5
A1M0SEWUJYX9K0	589	0	40	0	0	0	0	-75
A2KBTYNGUFRB9N	589	0	0	0	0	30	60	-20
A3POYFULMTNW1H	589	35	35	0	0	10	0	-10
ARQ4J4TLTPBNC	589	0	0	0	0	0	0	0
A1AVJRFM6L0RN8	588	0	0	0	0	0	0	0
ADAGUJNWMEPT6	588	0	0	0	0	0	10	0
A1LY3NJTYW9TFF	588	0	0	0	0	0	0	0
A14WWG6NKBDWGP	588	0	0	0	0	0	0	0
A1VYRD3HO2WDUN	588	0	0	0	0	0	0	0
A1XUURRBT9RYFW	588	0	0	0	0	0	0	0

Continued on next page

worker_id	task_id	anger_judg	disgust_judg	fear_judg	joy_judg	sadness_judg	surprise_judg	valence_judg
A1M0SEWUJYX9K0	588	0	0	0	0	0	0	0
A2KBTYNGUFRB9N	588	0	0	0	50	0	100	75
A3POYFULMTNW1H	588	0	0	0	10	0	20	10
ARQ4J4TLTPBNC	588	0	0	0	0	0	0	0
A1AVJRFM6L0RN8	599	0	0	0	0	0	0	0
ADAGUJNWMEPT6	599	0	0	5	0	0	0	1
A1LY3NJTYW9TFF	599	0	0	0	0	0	0	0
A14WWG6NKBDWGP	599	0	0	10	10	0	0	0
A1VYRD3HO2WDUN	599	0	0	0	0	0	0	0
A1XUURRBT9RYFW	599	0	0	0	0	0	0	0
A1M0SEWUJYX9K0	599	0	0	0	0	0	0	0
A2KBTYNGUFRB9N	599	0	0	0	5	0	0	0
A3POYFULMTNW1H	599	2	0	0	0	5	5	0
ARQ4J4TLTPBNC	599	40	50	40	0	100	40	0
A1AVJRFM6L0RN8	598	25	25	52	0	25	0	50
ADAGUJNWMEPT6	598	0	0	10	0	0	10	-10
A1LY3NJTYW9TFF	598	0	0	0	15	0	0	15
A14WWG6NKBDWGP	598	0	30	20	20	0	0	10
A1VYRD3HO2WDUN	598	0	0	0	0	0	0	0
A1XUURRBT9RYFW	598	0	0	0	0	0	0	0
A1M0SEWUJYX9K0	598	0	0	0	10	0	10	40
A2KBTYNGUFRB9N	598	0	0	5	50	0	0	25
A3POYFULMTNW1H	598	0	0	50	3	0	3	20
ARQ4J4TLTPBNC	598	0	0	0	0	0	0	0
A1AVJRFM6L0RN8	587	0	0	0	75	0	75	0
ADAGUJNWMEPT6	587	0	0	0	5	0	15	7
A1LY3NJTYW9TFF	587	0	0	10	0	0	30	15
A14WWG6NKBDWGP	587	0	0	0	20	0	30	30
A1VYRD3HO2WDUN	587	0	0	0	0	0	0	20
A1XUURRBT9RYFW	587	0	0	0	5	0	0	5
A1M0SEWUJYX9K0	587	0	0	0	20	0	25	60
A2KBTYNGUFRB9N	587	0	0	0	60	0	20	40

Bibliography

Stanley C. Ahalt, Ashok K. Krishnamurthy, Prakoon Chen, and Douglas E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3(3):277–290, 1990. ISSN 08936080. doi: 10.1016/0893-6080(90)90071-R.

Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying Support Vector Machines to Imbalanced Datasets. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Jean-Francois Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Machine Learning: ECML 2004*, volume 3201, pages 39–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-23105-9 978-3-540-30115-8. doi: 10.1007/978-3-540-30115-8_7. Series Title: Lecture Notes in Computer Science.

John Aldrich and others. RA Fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, 12(3):162–176, 1997.

Felipe Almeida and Geraldo Xexeo. Word Embeddings: A Survey. 2019.

David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via Dirichlet Forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, Montreal, Quebec, Canada, 2009. ACM Press. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553378.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A Simple but Tough-to-beat Baseline for Sentence Embeddings. 2017.

Hagai Attias. A variational bayesian framework for graphical models. In *In Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press, 2000.

A. Augustin and M. Venanzi. MBCC, 2017. https://github.com/alexandry-augustin/mbcc/.

Michael Bacharach. Normal Bayesian Dialogues. *Journal of the American Statistical Association*, 74(368):837, 1979. ISSN 01621459. doi: 10.2307/2286408.

Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. A Review of Machine Learning Algorithms for Text-Documents Classification. *Journal of Advances in Information Technology*, 1(1), 2010. ISSN 1798-2340. doi: 10.4304/jait.1.1.4-20.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. 2014.

Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07*, page 787, Amsterdam, The Netherlands, 2007. ACM Press. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277909.

Marco Baroni, Georgiana Dinu, and German Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1023.

Jon Atli Benediktsson and Philip H. Swain. Consensus theoretic classification methods. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(4):688–704, 1992.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. Automatic labelling of topics with neural embeddings. 2016.

William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556. Association for Computational Linguistics, 2012.

David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

David M. Blei and John D. Lafferty. A correlated topic model of Science. *The Annals of Applied Statistics*, 1(1):17–35, 2007. ISSN 1932-6157. doi: 10.1214/07-AOAS114.

David M. Blei and John D. Lafferty. Visualizing Topics with Multi-Word Expressions. 2009.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. 2016.

Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. Learning linear transformations between counting-based and prediction-based word embeddings. *PLOS ONE*, 12(9):e0184544, 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0184544.

Jonathan Bragg, Daniel S. Weld, and others. Crowdsourcing multi-label classification for taxonomy creation. In *First AAAI conference on human computation and crowdsourcing*, 2013.

Alexa Breuing and Ipke Wachsmuth. Let's Talk Topically with Artificial Agents! - Providing Agents with Humanlike Topic Awareness in Everyday Dialog Situations:. In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence*, pages 62–71, Vilamoura, Algarve, Portugal, 2012. SciTePress - Science and and Technology Publications. ISBN 978-989-8425-95-9 978-989-8425-96-6. doi: 10.5220/0003745900620071.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. A Novel Neural Topic Model and Its Supervised Extension. In *AAAI*, pages 2210–2216, 2015.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Improving Multi-Document Summarization via Text Classification. page 7, 2017.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-Graber, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296, 2009.

Hao Chen and Susan Dumais. Bringing order to the Web: automatically categorizing search results. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '00*, pages 145–152, The Hague, The Netherlands, 2000. ACM Press. ISBN 978-1-58113-216-8. doi: 10.1145/332040.332418.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179.

Bertrand Clarke. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *The Journal of Machine Learning Research*, 4:683–712, 2003.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. 2017.

Guido Consonni, Dimitris Fouskakis, Brunero Liseo, and Ioannis Ntzoufras. Prior Distributions for Objective Bayesian Analysis. *Bayesian Analysis*, 13(2):627–679, 2018. ISSN 1936-0975. doi: 10.1214/18-BA1103.

Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. page 25, 1995.

Marina Danilevsky, Chi Wang, Nihit Desai, Xiang Ren, Jingyi Guo, and Jiawei Han. Automatic Construction and Ranking of Topical Keyphrases on Collections of Short Documents. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 398–406. Society for Industrial and Applied Mathematics, 2014. ISBN 978-1-61197-344-0. doi: 10.1137/1.9781611973440.46.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 795–804, 2015.

A. P. Dawid and A. M. Skene. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied Statistics*, 28(1):20, 1979. ISSN 00359254. doi: 10.2307/2346806.

Cedric De Boom, Steven Van Canneyt, Steven Bohez, Thomas Demeester, and Bart Dhoedt. Learning Semantic Similarity for Very Short Texts. *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1229–1234, 2015. doi: 10.1109/ICDMW.2015.86.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. 41(6):391, 1990.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S. Bernstein, Alex Berg, and Li Fei-Fei. Scalable multi-label annotation. pages 3099–3102. ACM Press, 2014. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557011.

L. Devroye. The Equivalence of Weak, Strong and Complete Convergence in L1 for Kernel Density Estimates. *The Annals of Statistics*, pages 896–904, 1983.

Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. Multi-View Learning of Word Embeddings via CCA. page 9, 2011.

Emanuele Di Buccio, Massimo Melucci, and Federica Moro. Detecting verbose queries and improving information retrieval. *Information Processing & Management*, 50(2): 342–360, 2014. ISSN 03064573. doi: 10.1016/j.ipm.2013.09.003.

J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. 39(4):677–691, 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2599174.

Lei Duan, Satoshi Oyama, Haruhiko Sato, and Masahito Kurihara. Separate or joint? Estimation of multiple labels from crowdsourced annotations. *Expert Systems with Applications*, 41(13):5723–5732, 2014. ISSN 09574174. doi: 10.1016/j.eswa.2014.03.048.

Carsten Eickhoff. Cognitive Biases in Crowdsourcing. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18*, pages 162–170, Marina Del Rey, CA, USA, 2018. ACM Press. ISBN 978-1-4503-5581-0. doi: 10.1145/3159652.3159654.

Fatma El-Ghannam and Tarek El-Shishtawy. Multi-Topic Multi-Document Summarizer. *International Journal of Computer Science and Information Technology*, 5(6):77–90, 2013. ISSN 09754660, 09753826. doi: 10.5121/ijcsit.2013.5606.

Jeffrey L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

H.J. Escalante, C.A. Hernandez, J.A. Gonzalez, A. Lopez-Lopez, M. Montes, Eduardo F Morales, L Enrique Sucar, L. Villasenor, and M. Grubinger. The segmented and annotated IAPR TC-12 benchmark. *Computer Vision and Image Understanding*, 114 (4):419–428, 2010.

Kevin R Farrell and Richard J Mammone. Data fusion techniques for speaker recognition. In *Modern methods of speech processing*, pages 279–297. Springer, 1995.

Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

Eibe Frank and Remco R. Bouckaert. Naive bayes for text classification with unbalanced classes. In *Knowledge Discovery in Databases: PKDD 2006*, pages 503–510. Springer, 2006.

Nial Friel and Jason Wyse. Estimating the evidence-a review. *Statistica Neerlandica*, 66 (3):288–308, 2012.

Andrea Frome, Greg S Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. pages 2121–2129, 2013.

Cheryl Geisler, Charles Bazerman, Stephen Doheny-Farina, Laura Gurak, Christina Haas, Johndan Johnson-Eilola, David S Kaufer, Andrea Lunsford, and Carolyn R Miller. IText: Future directions for research on the relationship between information technology and writing. *Journal of Business and Technical Communication*, 15(3): 269–308, 2001.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

Christian Genest and James V. Zidek. Combining Probability Distributions: A Critique and an Annotated Bibliography. *Statistical Science*, 1(1):114–135, 1986. doi: 10.1214/ss/1177013825.

Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1734–1748, 2016.

Alex Gittens, Dimitris Achlioptas, and Michael W. Mahoney. Skip-Gram - Zipf + Uniform = Vector Additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1007.

Josef Goppert and Wolfgang Rosenstiel. Self-organizing maps vs. backpropagation: An experimental study. *Proc. of work. design methodol. microelectron. signal process*, pages 153–162, 1993.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

Derek Greene, Derek O'Callaghan, and Padraig Cunningham. How Many Topics? Stability Analysis for Topic Models. In Toon Calders, Floriana Esposito, Eyke Hullermeier, and Rosa Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 8724, pages 498–513. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-662-44847-2 978-3-662-44848-9. doi: 10.1007/978-3-662-44848-9_32.

Antonio Grieco, Massimo Pacella, and Marzia Blaco. On the Application of Text Clustering in Engineering Change Process. 62:187–192, 2017. ISSN 22128271. doi: 10.1016/j.procir.2016.06.019.

Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. Topic-based Evaluation for Conversational Bots. 2018.

Khaled M. Hammouda, Diego N. Matute, and Mohamed S. Kamel. CorePhrase: Keyphrase Extraction for Document Clustering. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Petra Perner, and Atsushi Imiya, editors, *Machine Learning and Data Mining in Pattern Recognition*, volume 3587, pages 265–274. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-26923-6 978-3-540-31891-0. doi: 10.1007/11510888_26.

G.H. Hardy, J.E. Littlewood, and G. Polya. *Inequalities.* Cambridge Mathematical Library. Cambridge University Press, 1952. ISBN 978-0-521-35880-4.

Hilda Hardy, Nobuyuki Shimizu, Tomek Strzalkowski, Liu Ting, Xinyang Zhang, and G. Bowden Wise. Cross-document summarization by concept classification. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02*, page 121, Tampere, Finland, 2002. ACM Press. ISBN 978-1-58113-561-9. doi: 10.1145/564376.564399.

Zellig S. Harris. Distributional Structure. *WORD*, 10(2-3):146–162, 1954. ISSN 0043-7956, 2373-5112. doi: 10.1080/00437956.1954.11659520.

Robert Herzog, Daniel Mewes, Michael Wand, Leonidas Guibas, and Hans-Peter Seidel. LeSSS: Learned Shared Semantic Spaces for Relating Multi-Modal Representations of 3d Shapes. *Computer Graphics Forum*, 34(5):141–151, 2015. ISSN 01677055. doi: 10.1111/cgf.12703.

Tom Heskes. Selecting weighting factors in logarithmic opinion pools. *Advances in neural information processing systems*, pages 266–272, 1998.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California, 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1162.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. 2012.

Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. 9(8):1735–1780, 1997.

Thomas Hofmann. Probabilistic latent semantic indexing. pages 50–57. ACM, 1999.

J.G. Hollands and I. Spence. Judgments of change and proportion in graphical perception. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 34 (3):313–334, 1992.

Timo Honkela. Self-organizing maps of words for natural language processing applications. In *Proceedings of the International ICSC Symposium on Soft Computing*, pages 401–407, 1997.

Jeff Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business.* Crown Business, 2009. ISBN 0-307-39621-5.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. *Machine Learning*, 95(3):423–469, 2014. ISSN 0885-6125, 1573-0565. doi: 10.1007/s10994-013-5413-0.

Weijing Huang. PhraseCTM: Correlated Topic Modeling on Phrases within Markov Random Fields. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, page 6, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daume III. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1162.

Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.212.

Ashish Jaiswal and Nitin Janwe. Hierarchical Document Clustering: A Review. In *2nd National Conference on Information and Communication Technology (NCICT)*, 2011.

Peng Jin, Yue Zhang, Xingyuan Chen, and Yunqing Xia. Bag-of-Embeddings for Text Classification. In *IJCAI*, pages 2824–2830, 2016.

Kristiina Jokinen, Antti Kerminen, Mauri Kaipainen, Tommi Jauhiainen, Graham Wilcock, Markku Turunen, Jaakko Hakulinen, Jukka Kuusisto, and Krista Lagus. Adaptive dialogue systems - interaction with interact. In *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue -*, volume 2, pages 64–73, Philadelphia, Pennsylvania, 2002. Association for Computational Linguistics. doi: 10.3115/1118121.1118131.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2): 183–233, 1999.

Joseph M. Kahn. A generative Bayesian model for aggregating experts' probabilities. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 301–308. AUAI Press, 2004.

Deb Kalyanmoy. *Multi-objective Optimization.* Search Methodologies. Springer, boston, ma edition, 2014.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

Alan F. Karr. *Probability (Springer Texts in Statistics).* Springer, 1993. ISBN 0387940715.

Chandra Khatri, Rahul Goel, Behnam Hedayatnia, Angeliki Metanillou, Anushree Venkatesh, Raefer Gabriel, and Arindam Mandal. Contextual Topic Modeling For Dialog Systems. 2018.

Douwe Kiela and Stephen Clark. Learning Neural Audio Embeddings for Grounding Semantics in Auditory Perception. *Journal of Artificial Intelligence Research*, 60: 1003–1030, 2017. ISSN 1076-9757. doi: 10.1613/jair.5665.

Hyun-Chul Kim and Zoubin Ghahramani. Bayesian Classifier Combination. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 11, La Palma, Canary Islands, Spain, 2012.

Seokhwan Kim, Rafael E. Banchs, and Haizhou Li. Wikipedia-based kernels for dialogue topic tracking. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 131–135. IEEE, 2014.

Seokhwan Kim, Rafael Banchs, and Haizhou Li. Exploring convolutional and recurrent neural networks in sequential labelling for dialogue topic tracking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 2016.

Yoon Kim. Convolutional neural networks for sentence classification. 2014.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014.

Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302, 2015.

Kerenaftali Klein, Stefanie Hennig, and Sanjoy Ketan Paul. A Bayesian Modelling Approach with Balancing Informative Prior for Analysing Imbalanced Data. *PLOS ONE*, 11(4):e0152700, 2016. ISSN 1932-6203. doi: 10.1371/journal.pone.0152700.

Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

Teuvo Kohonen. Self-organization of very large document collections: State of the art. In *ICANN 98*, pages 65–74. Springer, 1998.

Teuvo Kohonen. Essentials of the self-organizing map. 37:52–65, 2013. ISSN 08936080. doi: 10.1016/j.neunet.2012.09.018.

Daphne Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009. ISBN 978-0-262-01319-2.

Kundan Krishna and Balaji Vasan Srinivasan. Generating Topic-Oriented Summaries Using Neural Attention. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1697–1705, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1153.

Da Kuang, Jaegul Choo, and Haesun Park. Nonnegative Matrix Factorization for Interactive Topic Modeling and Document Clustering. In M. Emre Celebi, editor, *Partitional Clustering Algorithms*, pages 215–243. Springer International Publishing, Cham, 2015. ISBN 978-3-319-09258-4 978-3-319-09259-1. doi: 10.1007/978-3-319-09259-1_7.

Krista Lagus and Jukka Kuusisto. Topic identification in natural language dialogues using neural networks. In *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue* -, volume 2, pages 95–102, Philadelphia, Pennsylvania, 2002. Association for Computational Linguistics. doi: 10.3115/1118121.1118135.

Jouko Lampinen and Erkki Oja. Clustering properties of hierarchical self-organizing maps. 2(2):261–272, 1992.

Thomas K Landauer. On the computational basis of learning and cognition: Arguments from LSA. In *Psychology of Learning and Motivation*, volume 41, pages 43–84. Elsevier, 2002. ISBN 978-0-12-543341-9. doi: 10.1016/S0079-7421(02)80004-4.

Ian Lane, Tatsuya Kawahara, Tomoko Matsui, and Satoshi Nakamura. Out-of-Domain Utterance Detection Using Classification Confidences of Multiple Topics. 15(1):150–161, 2007. ISSN 1558-7916. doi: 10.1109/TASL.2006.876727.

Pierre Simon Laplace. Memoir on the probability of the causes of events. *Statist. Sci.*, 1 (3):364–378, 1986. doi: 10.1214/ss/1177013621.

Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1536–1545. Association for Computational Linguistics, 2011.

Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *ICML*, volume 14, pages 1188–1196, 2014.

Edward E. Leamer. *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. Wiley, 1978. ISBN 0-471-01520-2.

Remi Lebret and Ronan Collobert. Word Embeddings through Hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–490, Gothenburg, Sweden, 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-1051.

Yann LeCun, Patrick Haffner, Leon Bottou, and Yoshua Bengio. Object Recognition with Gradient-Based Learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–. Springer-Verlag, 1999. ISBN 3-540-66722-9.

Tak Yeon Lee, Alison Smith, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies*, 105:28–42, 2017. ISSN 10715819. doi: 10.1016/j.ijhcs.2017.03.007.

Omer Levy and Yoav Goldberg. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2050.

Kaican Li. Convergence rate of Gibbs sampler and its application. *Science in China Series A*, 48(10):1430, 2005. ISSN 1006-9283. doi: 10.1360/02ys0013.

Shuangyin Li, Yu Zhang, Rong Pan, Mingzhi Mao, and Yang Yang. Recurrent Attentional Topic Model. In *AAAI*, volume 17, pages 3223–3229, 2017.

Rensis Likert. *A Technique for the Measurement of Attitudes*. PhD thesis, 1932.

D. V. Lindley, A. Tversky, and R. V. Brown. On the Reconciliation of Probability Assessments. *Journal of the Royal Statistical Society. Series A (General)*, 142(2):146, 1979. ISSN 00359238. doi: 10.2307/2345078.

Robert V. Lindsey, William P. Headden III, and Michael J. Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*, pages 214–222. Association for Computational Linguistics, 2012.

Nick Littlestone and Manfred K. Warmuth. The Weighted Majority Voting Algorithm. *IEEE*, pages 256–261, 1989.

Lin Liu, Lin Tang, Wen Dong, Shaowen Yao, and Wei Zhou. An overview of topic modeling and its current applications in bioinformatics. 5(1), 2016-12. ISSN 2193-1801. doi: 10.1186/s40064-016-3252-8.

Qian Liu, Heyan Huang, Yang Gao, Xiaochi Wei, Yuxin Tian, and Luyang Liu. Task-oriented Word Embedding for Text Classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, page 10, 2018.

Yuan-Chao Liu, Ming Liu, and Xiao-Long Wang. Application of Self-Organizing Maps in Text Clustering: A Review. In Magnus Johnsson, editor, *Applications of Self-Organizing Maps*. InTech, 2012. ISBN 978-953-51-0862-7. doi: 10.5772/50618.

Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. 2018.

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996. ISSN 0743-3808, 1532-5970. doi: 10.3758/BF03204766.

Minnan Luo, Feiping Nie, Xiaojun Chang, Yi Yang, Alexander G Hauptmann, and Qinghua Zheng. Probabilistic Non-negative Matrix Factorization and its Robust Extensions for Topic Modeling. page 7, 2017.

Wenhan Luo, Bjorn Stenger, Xiaowei Zhao, and Tae-Kyun Kim. Automatic Topic Discovery for Multi-object Tracking. In *Proceedings of the 2015 Association for the Advancement of Artificial Intelligence (AAAI)*, page 7, 2015.

David J C MacKay. A Practical Bayesian Framework for Backprop Networks. page 11, 1992.

Kelsey MacMillan and James D. Wilson. Topic supervised non-negative matrix factorization. 2017.

Davide Magatti, Silvia Calegari, Davide Ciucci, and Fabio Stella. Automatic Labeling of Topics. pages 1227–1232. IEEE, 2009. ISBN 978-1-4244-4735-0. doi: 10.1109/ISDA.2009.165.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to information retrieval*. Cambridge University Press, New York, 2008. ISBN 978-0-521-86571-5.

N. Manukyan, M. J. Eppstein, and D. M. Rizzo. Data-Driven Cluster Reinforcement and Visualization in Sparsely-Matched Self-Organizing Maps. *IEEE Transactions on Neural Networks and Learning Systems*, 23(5):846–852, 2012. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2012.2190768.

Stephen McGregor, Kat Agres, Matthew Purver, and Geraint A. Wiggins. From Distributional Semantics to Conceptual Spaces: A Novel Computational Method for Concept Creation. *Journal of Artificial General Intelligence*, 6(1):55–86, 2015. ISSN 1946-0163. doi: 10.1515/jagi-2015-0004.

Kathleen R McKeown, Judith L Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. Towards Multidocument Summarization by Reformulation: Progress and Prospects. In *Proceedings of the 16th National Conference of the American Association for Artificial Intelligence (AAAI)*, pages 453– 460, 1999.

Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 490–499. ACM, 2007.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH*, pages 1045–1048. ISCA, 2010.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. 2013.

David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing Semantic Coherence in Topic Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, page 11, Edinburgh, United Kingdom, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep Learning Based Text Classification: A Comprehensive Review. 2020.

T. Minka, J.M. Winn, J.P. Guiver, S. Webster, Y. Zaykov, B. Yangel, A. Spengler, and J. Bronskill. *Infer.NET 2.6*. 2014. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

Thomas P. Minka. Bayesian model averaging is not model combination. pages 1–2, 2000.

Thomas P. Minka and John Winn. Gates. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2009.

Jeff Mitchell and Mirella Lapata. Vector-based Models of Semantic Composition. page 9, 2008.

Jeff Mitchell and Mirella Lapata. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429, 2010. ISSN 03640213. doi: 10.1111/j.1551-6709.2010.01106.x.

Marie-Francine Moens, Caroline Uyttendaele, and Jos Dumortier. Abstracting of legal cases: The potential of clustering based on the selection of representative objects. *Journal of the American Society for Information Science*, page 11, 1999.

Kristine Monteith, James L. Carroll, Kevin Seppi, and Tony Martinez. Turning Bayesian model averaging into Bayesian model combination. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2657–2663. IEEE, 2011.

Howard R. Moskowitz. Magnitude estimation: notes on what, how, when, and why to use it. *Journal of Food Quality*, 1(3):195–227, 1977. ISSN 0146-9428, 1745-4557. doi: 10.1111/j.1745-4557.1977.tb00942.x.

Robert Munro. *Human-in-the-Loop Machine Learning.* Spring, 2020. ISBN 978-1-61729-674-1.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 2901. NIH Public Access, 2015.

Ani Nenkova and Kathleen McKeown. A Survey of Text Summarization Techniques. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 43–76. Springer US, Boston, MA, 2012. ISBN 978-1-4614-3222-7 978-1-4614-3223-4.

Michael Nokel and Natalia Loukachevitch. Topic Models: Accounting Component Structure of Bigrams. page 8, 2015.

Manfred Opper and David Haussler. Calculation of the learning curve of Bayes optimal classification algorithm for learning a perceptron with noise. In *Annual Workshop on Computational Learning Theory: Proceedings of the fourth annual workshop on Computational learning theory*, volume 5, pages 75–87, 1991.

Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2): 111–126, 1994. ISSN 11804009, 1099095X. doi: 10.1002/env.3170050203.

Scott E. Page. *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies.* Princeton Univ. Press, Princeton, NJ, 3. printing, 1. pbk. printing edition, 2007. ISBN 978-0-691-13854-1 978-0-691-12838-2. OCLC: 836980749.

Alexandros Papangelis, Panagiotis Papadakos, Margarita Kotti, Yannis Stylianou, Yannis Tzitzikas, and Dimitris Plexousakis. LD-SDS: Towards an Expressive Spoken Dialogue System based on Linked-Data. 2017.

Giorgio Parisi. *Statistical field theory.* Addison-Wesley, 1988.

Tim Pearce, Mohamed Zaki, Alexandra Brintrup, Nicolas Anastassacos, and Andy Neely. Uncertainty in Neural Networks: Bayesian Ensembling. 2018.

Jiawei Han Jian Pei. Mining Frequent Patterns without Candidate Generation. page 12, 2004.

Ding Peng, Dai Guilan, and Zhang Yong. Contextual-LDA: A Context Coherent Latent Topic Model for Mining Large Corpora. pages 420–425. IEEE, 2016. ISBN 978-1-5090-2179-6. doi: 10.1109/BigMM.2016.72.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Elizabeth Pennisi. Seeking life's bare (genetic) necessities. *Science*, 272(5265):1098–1099, 1996.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 2018.

Penny M. Pexman, Paul D. Siakaluk, and Melvin J. Yap. Introduction to the research topic meaning in mind: semantic richness effects in language processing. *Frontiers in Human Neuroscience*, 7, 2013. ISSN 1662-5161. doi: 10.3389/fnhum.2013.00723.

Steven Pinker. *The Sense of Style: The Thinking Person's Guide to Writing in the 21st Century.* 2014.

Joaquin Quinonero Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift.* MIT Press, 2008.

Milena Rabovsky, Werner Sommer, and Rasha Abdel Rahman. The time course of semantic richness effects in visual word recognition. *Frontiers in Human Neuroscience*, 6, 2012. ISSN 1662-5161. doi: 10.3389/fnhum.2012.00011.

Dragomir R Radev, Hongyan Jing, and Malgorzata Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceeding NAACL-ANLP-AutoSum '00 Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization*, volume 4, page 8, 2000.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

Carl Edward. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.* MIT Press, Cambridge, Mass., 2006. ISBN 978-0-262-18253-9 0-262-18253-X.

F. Ravat. *Collaborative Decision Making: Perspectives and Challenges (Frontiers in Artificial Intelligence and Applications)*. IOS Press, 2008. ISBN 1-58603-881-8.

Gabriel Recchia and Michael N. Jones. The semantic richness of abstract concepts. *Frontiers in Human Neuroscience*, 6, 2012. ISSN 1662-5161. doi: 10.3389/fnhum.2012.00315.

Douglas L T Rohde, Laura M Gonnerman, and David C Plaut. An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence. page 33, 2006.

Andreas Ruckle, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations. 2018.

Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

Kamal Sarkar. Sentence Clustering-based Summarization of Multiple Text Documents. *TECHNIA-International Journal of Computing Science and Communication Technologies*, 2(1):325–335, 2009.

Peter Schonhofen. Identifying Document Topics Using the Wikipedia Category Network. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pages 456–462, Hong Kong, China, 2006. IEEE. ISBN 978-0-7695-2747-5. doi: 10.1109/WI.2006.92.

Hannes Schulz, Jeremie Zumer, Layla El Asri, and Shikhar Sharma. A Frame Tracking Model for Memory-Enhanced Dialogue Systems. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 219–227, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2626.

Ozan Sener and Vladlen Koltun. Multi-Task Learning as Multi-Objective Optimization. page 12, Montreal, Canada, 2018.

Burr Settles. Active Learning Literature Survey. *University of Wisconsin–Madison*, page 67, 2010.

Aashish Sheshadri and Matthew Lease. Square: A benchmark for research on computing crowd consensus. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K. Reddy. Short-Text Topic Modeling via Non-negative Matrix Factorization Enriched with Local Word-Context Correlations. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, pages 1105–1114, Lyon, France, 2018. ACM Press. ISBN 978-1-4503-5639-8. doi: 10.1145/3178876.3186009.

Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2008. ISBN 0-521-89943-5.

Edwin Simpson. *Combined Decision Making with Multiple Agents.* PhD thesis, University of Oxford, 2014.

Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. Dynamic Bayesian Combination of Multiple Imperfect Classifiers. In *Bayesian Combination of Multiple Imperfect Decision Makers*, Intelligent System Reference Library. Springer, 2013.

Abhishek SinghRathore and Devshri Roy. Ontology based Web Page Topic Identification. *International Journal of Computer Applications*, 85(6):35–40, 2014. ISSN 09758887. doi: 10.5120/14849-3211.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast-but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Y Ng. Zero-Shot Learning Through Cross-Modal Transfer. pages 935–943, 2013.

Rupesh Kumar Srivastava, Klaus Greff, and Jurgen Schmidhuber. Highway Networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, 2015.

S. S. Stevens. *Psychophysics: Introduction to its Perceptual, Neural, and Social Prospects.* John Wiley, New York, 1975.

Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics, 2007.

Geman Stuart and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6:721–741, 1984. ISSN 0162-8828.

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation.* MIT press, 2012.

James Surowiecki. *The wisdom of crowds.* Anchor Books, New York, NY, 1. ed edition, 2005. ISBN 978-0-385-72170-7.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150.

Kui Tang. Personalized Emotion Classification with Latent Dirichlet Allocation. 2012.

Wei Tang and Matthew Lease. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR Workshop on Crowdsourcing for Information Retrieval*, pages 66–75, 2011.

Fei Tian, Bin Gao, Di He, and Tie-Yan Liu. Sentence Level Recurrent Topic Model: Letting Topics Speak for Themselves. 2016.

Sabrina Tiun, Rosni Abdullah, and Tang Enya Kong. Automatic Topic Identification Using Ontology Hierarchy. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Alexander Gelbukh, editors, *Computational Linguistics and Intelligent Text Processing*, volume 2004, pages 444–453. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-41687-6 978-3-540-44686-6. doi: 10.1007/3-540-44686-9_43.

Andrew Turpin, Falk Scholer, Stefano Mizzaro, and Eddy Maddalena. The Benefits of Magnitude Estimation Relevance Assessments for Information Retrieval Evaluation. pages 565–574. ACM Press, 2015. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767760.

Stefan Ultes, Lina M. Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Pawel Budzianowski, Nikola Mrksic, Tsung-Hsien Wen, Milica Gasic, and Steve Young. PyDial: A Multi-domain Statistical Dialogue System Toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-4013.

A. W. van der Vaart. *Asymptotic Statistics (Cambridge Series in Statistical and Probabilistic Mathematics).* Cambridge University Press, 2000. ISBN 0521784506.

Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 155–164. ACM Press, 2014. ISBN 9781450327442. doi: 10.1145/2566486.2567989.

Pavlos Vougiouklis, Jonathon Hare, and Elena Simperl. A neural network approach for knowledge-driven response generation. In *Proceedings of COLING 2016, the 26th*

International Conference on Computational Linguistics: Technical Papers, pages 3370–3380, 2016.

Jeroen Vuurens, Arjen P. de Vries, and Carsten Eickhoff. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR'11)*, pages 21–26, 2011.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112. ACM, 2009.

Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 697–702. IEEE, 2007.

A. J. M. M. Weijters. The BP-SOM architecture and learning rule. *Neural Processing Letters*, 2(6):13–16, 1995. ISSN 1370-4621, 1573-773X. doi: 10.1007/BF02309010.

AJMM Weijters, Antal van den Bosch, and H Jaap van den Herik. Behavioural aspects of combining backpropagation learning and self-organizing maps. *Connection Science*, 9(3):235–252, 1997.

Peter Welinder, Steve Branson, Pietro Perona, and Serge J. Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.

DRGHR Williams and GE Hinton. Learning representations by back-propagating errors. *Nature*, 323(6088):533–538, 1986.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised Deep Embedding for Clustering Analysis. In *Proceedings of the 33rd International Conference on Machine Learning*, page 10, New York City, 2016.

YunYun Yang, Lucy Akers, Thomas Klose, and Cynthia Barcelon Yang. Text mining and visualization tools – Impressions of emerging capabilities. *World Patent Information*, 30(4):280–293, 2008. ISSN 01722190. doi: 10.1016/j.wpi.2008.01.007.

Melvin J. Yap, Penny M. Pexman, Michele Wellsby, Ian S. Hargreaves, and Mark J. Huff. An Abundance of Riches: Cross-Task Comparisons of Semantic Richness Effects in Visual Word Recognition. *Frontiers in Human Neuroscience*, 6, 2012. ISSN 1662-5161. doi: 10.3389/fnhum.2012.00072.

Ahmet Yildirim, Suzan Uskudarli, and Arzucan Ozgur. Identifying Topics in Microblogs Using Wikipedia. *PLOS ONE*, 11(3):e0151885, 2016. ISSN 1932-6203. doi: 10.1371/journal.pone.0151885.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent Trends in Deep Learning Based Natural Language Processing. 2017.

Haijun Zhai, Jiafeng Guo, Qiong Wu, Xueqi Cheng, Huawei Sheng, and Jin Zhang. Query Classification Based on Regularized Correlated Topic Model. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 552–555, Milan, Italy, 2009. IEEE. ISBN 978-0-7695-3801-3. doi: 10.1109/WI-IAT.2009.91.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2016.

Weizhong Zhao, James J Chen, Roger Perkins, Zhichao Liu, Weigong Ge, Yijun Ding, and Wen Zou. A heuristic approach to determine an appropriate number of topics in topic modeling. *BMC Bioinformatics*, 16(S13), 2015. ISSN 1471-2105. doi: 10.1186/1471-2105-16-S13-S8.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27, 2015.