

Semantically Tagging Images of Landmarks

Heather S. Packer, Jonathon S. Hare, Sina Samangooei, and Paul H. Lewis

Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
{hp3|jsh2|ss|ph1}@ecs.soton.ac.uk

Abstract. Semantic tagging allows images to be linked with URIs from web resources. Disambiguating URIs which correspond with an image’s visual content is challenging. Previous work has largely failed to effectively contextualise the knowledge provided by the Semantic Web and the user-provided keyword tags in images. We propose an algorithm which uses geographical coordinates and keywords of similar images to recommend semantic tags that describe an image’s visual content. Our algorithm uses the semantic stores YAGO2, DBPedia and Geonames. These stores allow us to handle multi-lingual keyword tags and disambiguate between alternative names for landmarks.

1 Introduction

Image tagging systems predominately allow no explicit semantic links describing the visual content of images. The meaning of user assigned keyword tags can be ambiguous because they typically use only a few words. For example, a user may tag an image with “cathedral”. This keyword tag does not specify which cathedral. In order to improve the semantics of tags, “Semantic tags” (also known as “Machine tags”) have been used to add explicit context in the form of links and additional information such as geographic coordinates. Specifically, a semantic tag is a link defined by a Uniform Resource Identifier (URI) referencing an entity defined on the Semantic Web. Semantic tags enrich images by linking them to resources that provide more information about the entities contained within the image. A key challenge in semantic tagging is assigning the correct URIs.

This paper describes an approach that attempts to semantically enrich a query image, that has no information other than its pixel content, with URIs of the entities depicted in the image. Firstly, we identify visually similar images by comparing the visual features of the query image against a large corpus of partially annotated images. This corpus contains a mixture of images that have been geotagged and images with keyword tags, or images with a mixture of the two types of annotation. Using the geolocation information from the similar images we estimate the most likely geographic coordinates of the query. We then compare the raw keyword tags of the similar images to entities contained in the YAGO2 [1] knowledge-base and validate whether any geographic-relations of the entities are close to the query image’s estimated location using the DBPedia [2]

and Geonames [3] knowledge-bases. Finally, we use these selected entities to construct a list of URIs which are relevant to the content of the query image¹.

The broader motivation of this work is to bridge the gap between unstructured data attainable in real time, such as a GPS-location or image of a landmark, and, rich, detailed and structured information about that landmark, therefore facilitate more informed search and retrieval. For example, users can engage in semantic searching; targeting a particular church, architect, period in history or architectural style. This kind of structured search can also support event detection [4] or lifelogging [5] by helping collate documents which refer to a particular entity more accurately.

This work provides two contributions beyond state-of the-art. Specifically, we verify semantic tags using distance according to the height of the entities identified in an image, and we also recommend URIs using a multi-language tag index derived from multiple semantic web knowledge sources. The multi-lingual index allows us to make recommendations from keywords in foreign languages by identifying alternative names to utilise in our approach.

The structure of the paper is as follows: firstly, we discuss related work on tagging systems. Secondly we present details of our tagging approach. Finally, we discuss the use of our system with some examples and provide our conclusions.

2 Related Work

Automatic image annotation is widely studied, but techniques that integrate multiple contexts using semantic web resources are relatively rare. The following review looks at works that recommend tags using both geographical coordinates and visual image similarity. SpiritTagger [6] recommends keywords that reflect the *spirit* of a location. It finds visually similar images using colour, texture, edge and SIFT features [7], and clusters tags based within a geographical radius. These selected tags are ranked based on frequency and then importance. This enables their algorithm to recommend tags that are specific to a geographic region. The focus of our work is to recommend semantic tags (URIs) that describe a place of interest, not tags relating to a larger area.

Moxley et al. [8] use a corpus of images and their tags, and organise them into sets of places, events and visual components. These are clustered based on the co-occurrence of words and the distance between named geographical entities. If an image matches, the wikipedia title is used as a recommendation for the name of the landmark in the image. This approach uses limited information from wikipedia to identify and recommend tags. In our approach, we aim to validate semantic tags using additional information from semantic data sources.

Similar to Moxley et al. [6], Kennedy and Naaman [9]’s approach also considers the importance of tags relevant to a specific area or event. Their approach generates a representative image set for landmarks using image tags and geographic coordinates. Their technique identifies tags that occur frequently within

¹ A demonstration of our approach can be found here: <http://gtb.imageterrier.org>

one geographic area, but infrequently elsewhere, in order to identify tags that are uniquely local. They also filter tags that only occur during specific time ranges; this enables them to identify events such as the “New York Marathon” and determine whether this is relevant to a query image by analysing the date it was taken. Their focus was not concerned with recommending semantic tags.

There are a number of datasets that contain flickr images and related URIs. For instance, the Ookaboo dataset was manually created by 170,000 contributors who submitted images and classify them against a topic from wikipedia ². In contrast, our approach recommends URIs automatically for an untagged image, by using tags from images that share visual features. The flickrtm wrapper API allows users to search with a URI of an entity on Wikipedia and search for images that depict that entity. In particular, you can search for images within a user-specified distance of the geographical location of the searched entity (if the entity has a geographical location). This is the opposite problem to us, our query is an image depicting a landmark where the landmark is unknown, whereas their query is an entity on Wikipedia. The work by [10] identify entities using natural language processing by stemming words to find their root or base by removing any inflections, using Wordnet. They then identify any relationships between these entities using the hypernym, holonym, meronym, and toponym relationships described in Wordnet to create the triples describing the entities described in flickr tags. Our approach supports [10]’s, by generating the URI describing entities depicted in an image when it has no tags, so that their approach could generate triples.

A number of other approaches simply use location and visual features to annotate images (e.g. [11]). There has also been work to recommend tags, based on existing annotations (e.g. [12]), and recommending semantic entities, based on existing tags (e.g. [13]).

3 Approach

Our semantic tag recommendation approach has five steps. Firstly, we search a large index of visual features extracted from partially annotated images with the features extracted from a query image in order to find images similar to the query. The index contains images that have either geographic tags or keyword tags (or a mixture of the two). From the set of similar images with geographic locations we calculate a robust average of the latitude and longitude which estimates the geographic location of the query. Secondly, we use the keyword tags associated with the similar images to identify entities close to the estimated coordinates from YAGO2. Thirdly, we classify the types of entities that are possible recommendations using the type hierarchies of YAGO2, DBPedia and Geonames. In the fourth step, we restrict our recommendations based on their height and distance. In the final step, we expand our set of URIs with those from the closest city in order to try and identify additional relevant semantic entities.

² Ookaboo: <http://ookaboo.com/o/pictures/>

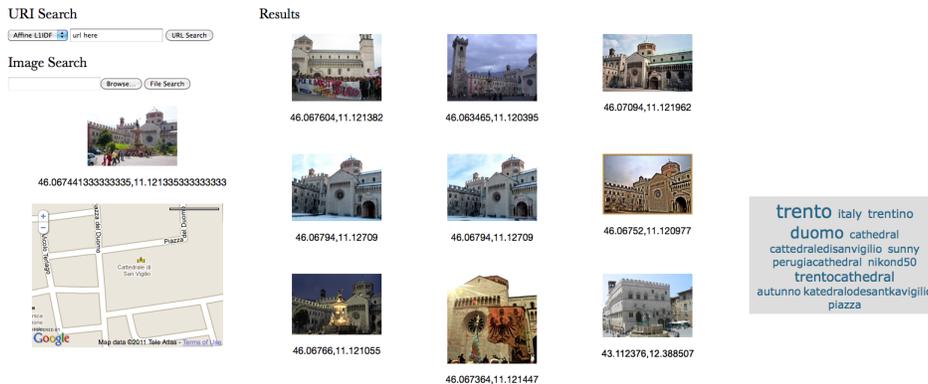


Fig. 1. The query image of Trento Cathedral, and the resulting images that matched, based on an index of SIFT features.

The approach has been developed using an index of images crawled from Flickr representing the Trentino region in Italy. More details of the dataset can be found in Section 4. In the remainder of this section, we walk through each of the five steps in detail.

3.1 Visual Image Similarity

Firstly, we compare the visual content of an untagged query with the visual content of each image in a dataset of tagged images. This is achieved by comparing via a BoVW (Bag of Visual Words) [14] representation of both query image and dataset images, extracted using the OpenIMAJ Toolkit³ [15]. For efficient retrieval, the BoVW of the dataset images is held in a compressed inverted index, constructed using ImageTerrier⁴ [16]. Once constructed, the index can be used to retrieve dataset images which are most visually similar to a query image; the tags and geographic locations of the closest dataset images are passed on to the next steps of our process. Specifically, the retrieval engine is tuned to only retrieve images that match with a very high confidence and thus only match the specific landmark/object in the query image; the aim is not to classify images into landmark classes, but rather to identify a specific instance.

The BoVW image representations are constructed by extracting difference-of-Gaussian SIFT features [17] from an image and quantising them to a discrete vocabulary. A vocabulary of 1 million features was learnt using approximate K-Means clustering [18] with SIFT features from the MIRFlickr25000 dataset [19]. Once the content of each image is represented as a set of visual terms, we construct an inverted index which encodes each term in an image. The inverted index is augmented with the orientation information of the SIFT feature corresponding to each term; this extra geometric information allows us to improve

³ OpenIMAJ Toolkit: <http://openimaj.org>

⁴ ImageTerrier: <http://imageterrier.org>

retrieval precision using an orientation consistency check [20] at query time. For every query image, the SIFT features are extracted and quantised. The set of visual terms form a query against the inverted index which is evaluated using the Inverse-Document-Frequency weighted L1 distance metric [21]. Using this strategy we select the top ten images in the dataset. These images provide us with potential keyword tags and geographic coordinates.

An iterative approach is used to robustly estimate the geographic location of the query from the set of geotagged result images. The technique needs to be robust as there is a high probability of outliers. Starting with all the matching geotagged images, a geographic centroid is found and the image which is geographically furthest from the centre is removed. The centroid is then updated with the remaining images. This process continues iteratively until the distance between the current centroid and furthest point from the centroid is less than a predefined threshold. Through initial tests on our dataset, we found that the threshold of 0.8 returned between 70% and 100% of images that were visually similar. An example of the visual similarity search and geographic localisation for a query image of Trento Cathedral is illustrated in Figure 1.

3.2 Keyword-Entity Matching

The second step aims to find URIs representing entities in the query image by attempting to match keyword tags to the names of entities. This can be problematic because it is common for keyword tags to contain more than one word without white space. Therefore, when searching for an entity in YAGO2 that matches a tag representing more than one word it will yield no matches. For example, the keyword tag ‘trentocathedral’ will not match the YAGO2 entity ‘Trento Cathedral’. In order to enable us to search for flattened tags, we performed a pre-processing step to create additional triples relating flattened tags to entities within YAGO2. We also flattened the entities relating to an entity through the “isCalled” property because it contains alternate terms used to refer to an instance (including foreign language names). For example, the YAGO2 entity for “Trento Cathedral” can also be called “Cattedrale di San Vigilio” and “Katedralo de Santka Vigilio”. Thus, we also use the flattened entity names “cattedraledisanvigilio” and “katedralodesantkavigilio” to represent “Trento Cathedral”. These additional triples and YAGO2 are used to look up all the tags using exact string matching. If there are matching entities then we check that they are in the same city (using the geographical coordinates from YAGO2 and the estimated coordinates from step one). In our Trento example, we retrieve the URIs shown in Table 1 from the image’s tags.

3.3 Cross-Source Category Matches

The aim of the third step is to determine whether the entities identified in step 2 and the keyword tags of the similar images are of a specific type. We organised these types into general categories, including town/city, region, country, date, weather, season, mountain, building, activity, transport and unknown. This list

Table 1. The tags and YAGO2 matches for the first four result images.

Image	Tags	YAGO2 Matches
1	cathedral, trento, duomo	http://mpii.de/yago/resource/Trento_Cathedral http://mpii.de/yago/resource/Cathedral http://mpii.de/yago/resource/Trento
2	trento, italy, trentino, duomo	http://mpii.de/yago/resource/Province_of_Trento http://mpii.de/yago/resource/Cathedral http://mpii.de/yago/resource/Trento http://mpii.de/yago/resource/Italy
3	cattedrale- disanvigilio, cathedral, trento	http://mpii.de/yago/resource/Trento_Cathedral http://mpii.de/yago/resource/Cathedral http://mpii.de/yago/resource/Trento
4	italia, autunno, perugiacathedral	http://mpii.de/yago/resource/Italy http://mpii.de/yago/resource/wordnet_fall_115236859 http://mpii.de/yago/resource/Perugia_Cathedral

was derived from a sample set of tags from our corpus of Flickr images from the Trentino region, and can be used to categorise 78% of all tags in the corpus. This categorisation allows us to search for entities that have a geographical location because we can filter out entities of type date, weather and activity which is not specific to one geographical location.

In order to categorise the identified matches we found in YAGO2 (from the previous step), we look up the identified entities in DBPedia and Geonames. This is possible because YAGO2 incorporates the property “hasWikipediaUrl” which DBPedia and Geonames both reference. In order to identify the matches’ categories we recurse through the type hierarchies of DBPedia and compare lexically the hierarchies. We also map the Geonames *feature code* which categorises towns, regions, countries, mountains and other landscape features, with our categories. We add any entities that we cannot categorise to the ‘unknown’ set.

In our Trento Cathedral example, we categorise the entities identified from YAGO2 (see the YAGO2 Matches in Table 1) and the tags from similar images (see the tag cloud shown in Figure 1). Table 2 shows the selected categories and corresponding properties that were used to infer these categories.

3.4 Geographic Distance Constraints

Using the categories assigned to the tags in step 3, we aim to verify whether the entities (identified in step 2) which have a geographical location, are located within a certain distance from the predicted location (from step 1). We predefine acceptable maximum distances for entities of type city, region and mountain and found that through preliminary testing that these were suitable values (see Table 3). It is possible to evaluate the height of buildings using the “heightStories” and “floorCount” properties in DBPedia. We approximate a viewing distance for buildings using these properties. Based on an empirical evaluation, with every

Table 2. Entities, their hierarchies and category.

Entity or Tag	Hierarchy	Category
Trento Cathedral	Cathedrals in Italy, Cathedral, Church, Place of Worship, Building	Building
Trento	Cities and towns in Trentino-Alto Adige, City	Town/City
Trentino	Provinces of Italy, State, District, Region	Region
Italy	Italian-speaking countries, Country	Country
Luglio (July)	Months, Calendar Month, Time Period	Date
Autunno (Autumn)	Season	Season
Sunny	Weather	Weather
Piazza	Public Square, Tract, Location	Place
perugiacathedral	Cathedrals In Italy, Cathedral, Church, Place of Worship, Building	Building
NikonD50	Nikon DSLR Camera, Camera, Photographic Equipment, Equipment, Artifact, Object, Physical Entity, Entity	Unknown
cattedraleisanvigilio	Cathedrals in Italy, Cathedral, Church, Place of Worship, Building	Building
katedralodesantkavigilio	Cathedrals in Italy, Cathedral, Church, Place of Worship, Building	Building

floor we estimate that it is possible to see a building from a further 5 meters away. This is, however, an approximation because it will differ with landscape features such as elevation, the average height of buildings around the building in the query image, and the height of the floors.

Our approach cannot guarantee that recommended entities are contained within the query image because an entity might be within range of the estimated location but it may not be within sight of the camera because other objects may block the view or recommended entities might be located in a different direction. However, we make our recommendation because images matched with step 1 contain reference to these entities. Therefore, we hypothesise that there is a high likelihood that these recommended entities are depicted in the query image.

Table 3. Maximum allowed distances.

Category	Maximum Distance (KM)
place	0.5
city	7
region	30
mountain	50

In our Trento Cathedral example, the entity Duomo of category building has 5 floors and is 10 meters from the estimated geolocation. Using our approach we validate that the Duomo is within our specified range. In the tags related to the

similar images, we identify that the building “Perugia Cathedral” has 6 floors and is 343.5 kilometers away from the estimated location. Therefore, we do not recommend URI of this building because its is not within range.

3.5 Recommendation Expansion

In the final step of our approach, we aim to derive further matches by expanding our search terms. Specifically, we expand all non place entities (excluding entities of the type city, town, region and country) with the the closest place name, using the pattern [place name][non place entity]. This allows us to disambiguate entities that are common to many places, such as town halls, police stations and libraries. We then check whether the matches are located close to the estimated coordinates. In our Trento Cathedral example, the tag piazza is expanded to “Trento piazza” which is linked to by YAGO2 by the “isCalled” property to the entity “Trento-Piazza_del_Duomo”. We then validate that the “Trento-Piazza_del_Duomo” is categorised with place and is within the geographic distance range of 0.5km from the estimated geographical location. In Table 4 we detail the extended tags which we attempt to match to YAGO2. Table 5 details the recommended URIs for our example.

Table 4. Extended Tags and URIs.

Extended tag [place][non place entity]	URI
Trento Cathedral	http://www.mpii.de/yago/resource/Trento_Cathedral
Trento Piazza	http://www.mpii.de/yago/resource/Trento_Piazza

Table 5. Recommended Entities and URIs.

Recommended Entity	URI
Province of Trento	http://en.wikipedia.org/wiki/Province_of_Trento
Trento	http://en.wikipedia.org/wiki/Trento
Italy	http://en.wikipedia.org/wiki/Italy
Trento Cathedral	http://www.mpii.de/yago/resource/Trento_Cathedral
Trento Piazza	http://www.mpii.de/yago/resource/Trento_Piazza

4 Examples and Discussion

In this section, we discuss the recommended URIs for four example query images. The dataset of partially annotated images used as the basis to testing the approach was crawled from Flickr. We first downloaded approximately 150,000 geo-tagged images that were within the bounds of the province of Trentino, an area of 2,397 square miles in the north of Italy. This set was then enriched with an additional 400,000 images with keyword tags relating to the Trentino region.

In total our image set consists of 472,565 images because of the intersection of these images sets⁵. We randomly sampled 3,250 images from this set and manually identified the theme of the image (see Table 6).

Table 6. Sample of topics from 3,250 images

Theme	Percentage (%)
People	31.2
Landscape	26.2
Houses	16.5
Animals	12.5
Churches	5.8
Other	5.6
Transport	1.6
Trento Cathedral	0.6
Buonconsiglio Castle	0

We considered using standard image sets such as the European Cities 50K[22] dataset and MIRFlickr[19]. However, we wanted to include images from the surrounding area because often they are the most similar visually, areas typically have a particular style due to tradition and age of the area. The European cities set contains images from different cities and does not include whole regions. Similarly we chose not to use the MIRFlickr image set as it was not suitable because our approach requires both tags to identify landmarks and geotags to disambiguate between landmarks, and 88.948% of the images in MIRFlickr did not contain geotags. Whereas our image set contains over double the number of geotagged images at 23% compared to 11%. To the best of our knowledge there were no suitable datasets which contained a complete area of geotagged images, or that contained a ground truth of URIs associated with each image.

4.1 Example 1

The first query image depicts Trento Cathedral, the Neptune Fountain and a plaza, and the visual feature search returned the images that depict the cathedral (see Figure 2). The seventh similar image is a photograph of Trento Cathedral and that image is not geographically close to the landmark’s actual location. While estimating the geographical location of the image, the seventh image’s geotags are removed by the threshold described in Section 3.1, and therefore does not effect the recommended URIs. Our approach correctly identifies that the query image contains Trento Cathedral and recommends the URIs for the Italian and English wikipedia URIs for the cathedral because the tag cloud contains ‘trentocathedral’, ‘trento’ and ‘cathedral’. It also recommends URIs relating to the region, such as town, region, and country, and URIs relating

⁵ Our image set can be downloaded here: <http://degas.ecs.soton.ac.uk/~hp07r/fullcollection.csv>

to ecclesiastical buildings, notably it recommended URIs about the cathedral's Pope (see following list).

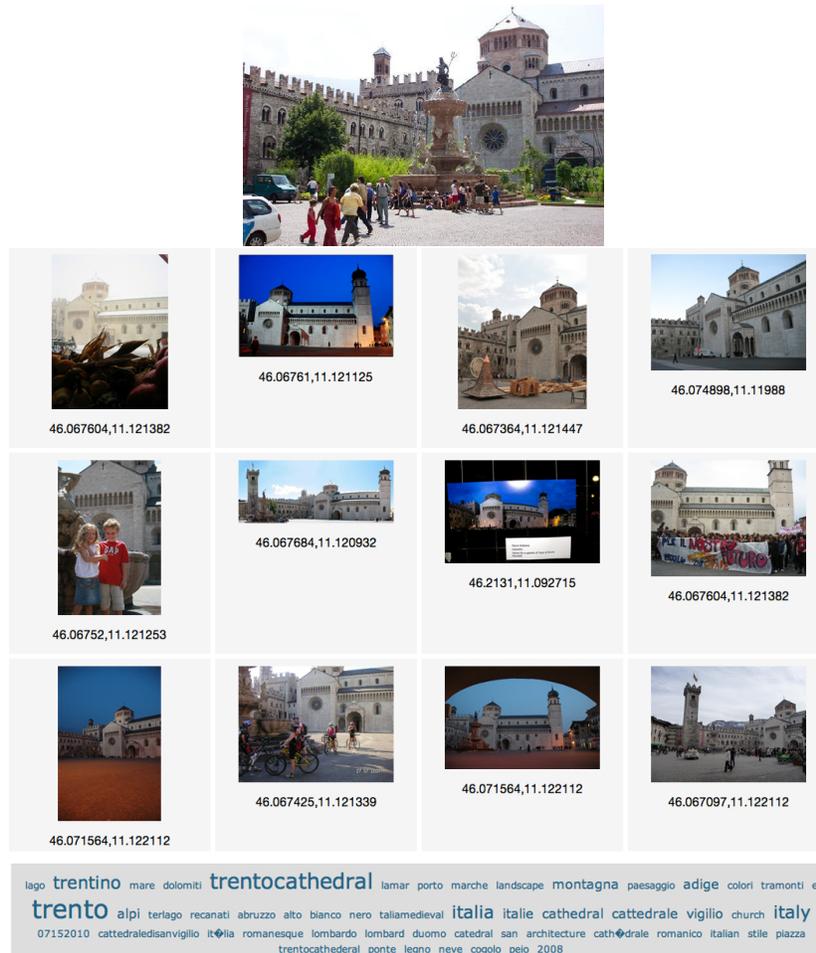


Fig. 2. The query image of Trento Cathedral, similar images and tag cloud.

1. http://en.wikipedia.org/wiki/Trento_Cathedral
2. http://it.wikipedia.org/wiki/Cattedrale_di_San_Vigilio
3. http://it.wikipedia.org/wiki/Provincia_autonoma_di_Trento
4. <http://www.mpii.de/yago/resource/Italy>
5. <http://en.wikipedia.org/wiki/Italy>
6. <http://en.wikipedia.org/wiki/Trento>
7. <http://it.wikipedia.org/wiki/Church>
8. <http://en.wikipedia.org/wiki/Alps>

9. <http://en.wikipedia.org/wiki/Cathedral>
10. <http://it.wikipedia.org/wiki/Trento>
11. <http://it.wikipedia.org/wiki/Cattedrale>
12. http://it.wikipedia.org/wiki/The_Church
13. http://en.wikipedia.org/wiki/Province_of_Trento
14. <http://en.wikipedia.org/wiki/Church>
15. http://www.mpii.de/yago/resource/Province_of_Trento
16. http://en.wikipedia.org/wiki/Pope_Vigilius
17. http://it.wikipedia.org/wiki/Papa_Vigilio
18. <http://it.wikipedia.org/wiki/Cathedral>
19. <http://en.wikipedia.org/wiki/Trentino>
20. <http://www.mpii.de/yago/resource/Trento>
21. <http://en.wikipedia.org/wiki/Mountain>
22. <http://en.wikipedia.org/wiki/Alto>

4.2 Example 2

The second query image depicts Trento Cathedral, and the visual feature search correctly matched seven images of the cathedral (see Figure 3). From the tag cloud we can see that one or more of the similar images has been incorrectly tagged with ‘Buonconsiglio’ and ‘Castle’. These tags refer to Buonconsiglio Castle which is approximately 700 meters from Trento Cathedral. In step four of our approach, we disambiguate between places of interest when there is a distance greater than 0.5km. However, in this case, our approach was unable to disambiguate between the two places of interest because all the geotagged images were within 0.5km of Trento Cathedral (as defined on Geonames) and contained tags relating to both the cathedral and castle. If the image tagged with ‘Buonconsiglio’ was geographically located at the castle, then our approach would have only recommended URIs relating to the cathedral. Our approach recommended the URIs in example 1 and those in the following list, and recommended URIs that relate to both Buonconsiglio Castle and Trento Cathedral.

1. http://it.wikipedia.org/wiki/Castello_del_Buonconsiglio
2. http://en.wikipedia.org/wiki/Castello_del_Buonconsiglio
3. <http://it.wikipedia.org/wiki/Castello>
4. http://www.mpii.de/yago/resource/Trento_Cathedral

4.3 Example 3

The third query image also depicts Trento Cathedral (see Figure 4). The visual feature search matched three images but only one of the images depicted Trento Cathedral. None of these images were tagged, therefore our approach could not find or expand any tags to look up entities in YAGO2, DBPedia or Geonames.

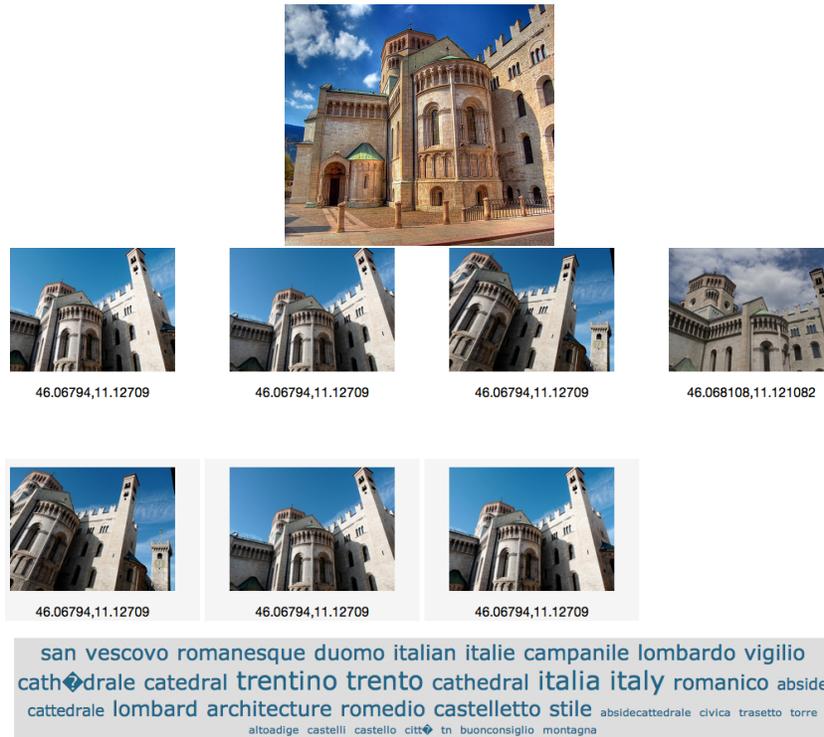


Fig. 3. The query image of Trento Cathedral, similar images and tag cloud.

4.4 Example 4

The fourth query image depicts Buonconsiglio Castle. The visual feature search returned over 20 images. Figure 5 shows the first eight images which are the most similar to the query image. The first eight images all depict the castle. The visual feature search also returned images of Trento Cathedral, hence the tag cloud contains tags about the cathedral: cathedral, catedral, cathdrale, catedrale, and vigilio. Unlike our second example, our approach was able to disambiguate between the castle and cathedral because the similar images were correctly geotagged within 0.5km from the photographed landmark. Our approach expanded the tag Buonconsiglio with castle (see Section 3.5), because it determined that castle was a type of building, and thus was able to identify the wikipedia URI http://en.wikipedia.org/wiki/Buonconsiglio_Castle. The following list contains our approach's recommended URIs.

1. http://en.wikipedia.org/wiki/Buonconsiglio_Castle
2. http://it.wikipedia.org/wiki/Castello_del_Buonconsiglio
3. http://en.wikipedia.org/wiki/Castello_del_Buonconsiglio
4. http://it.wikipedia.org/wiki/Provincia_autonoma_di_Trento
5. <http://en.wikipedia.org/wiki/Trento>

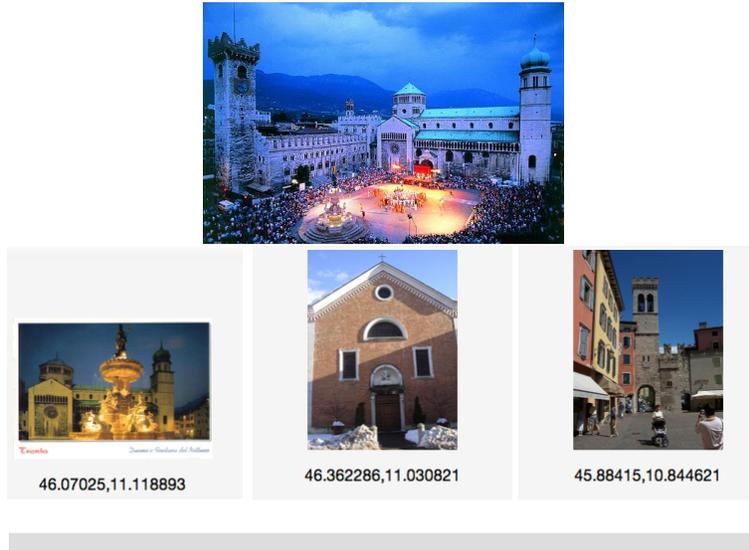


Fig. 4. The query image of Trento Cathedral, similar images and tag cloud.

6. <http://www.mpii.de/yago/resource/Italy>
7. <http://en.wikipedia.org/wiki/Italy>
8. <http://it.wikipedia.org/wiki/Castello>
9. <http://it.wikipedia.org/wiki/Trento>
10. <http://www.mpii.de/yago/resource/Trento>
11. <http://en.wikipedia.org/wiki/Trentino>
12. http://en.wikipedia.org/wiki/Province_of_Trento
13. http://www.mpii.de/yago/resource/Province_of_Trento

Our approach can be hindered by the quality of information in the semantic knowledge stores — the set of tags and the coordinates. The examples discussed in this section show that without approximately correct coordinates or tags, our algorithm will not be able to identify and recommend accurate semantic tags. Nor will our approach be able to validate the coordinates of places of interest, if the knowledge base does not contain them.

5 Conclusion

In this paper, we present an algorithm to recommend URIs that represent the visual content of an image and focus on identifying places of interest using geographical information from YAGO2, DBPedia, and Geonames. In order to use these knowledge sources, we use large-scale image matching techniques to find similar images that are then used to estimate geo-coordinates and potential tags.

The four examples show that the quality our results highly depends on the quality of the image matching techniques and the reference corpus. Our approach

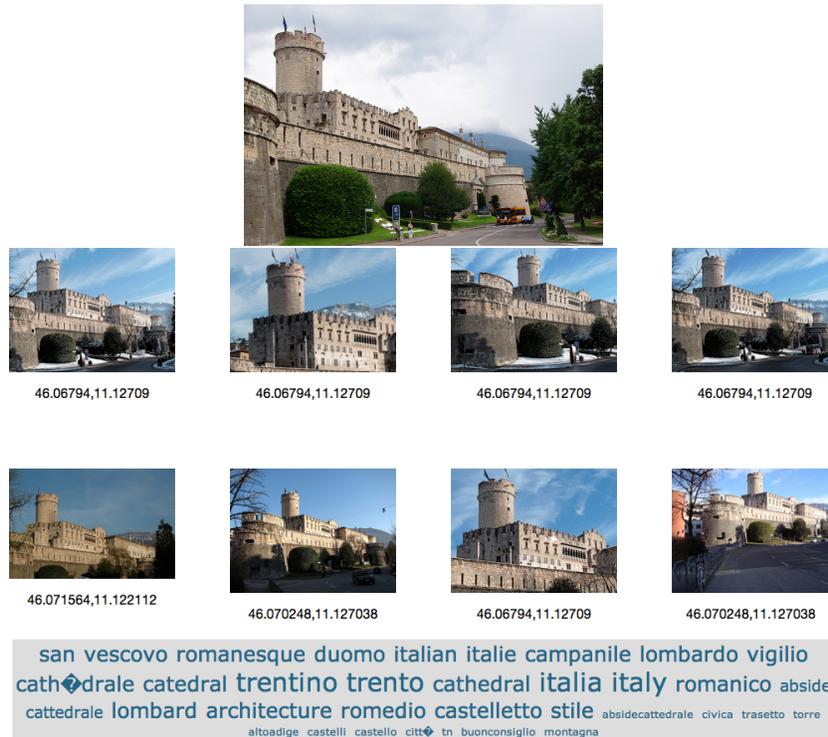


Fig. 5. The query image of Buonconsiglio Castle, similar images and tag cloud.

performs best when there are accurate tags and geotags and this is not always the case with collections of images. For future work, we plan to develop approaches that consider how to better handle keyword tags and geotags that are incorrect.

Acknowledgments

This work was funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements n° 270239 (ARCOMEM), 287863 (TrendMiner) and 231126 (LivingKnowledge) together with the LiveMemories project, graciously funded by the Autonomous Province of Trento (Italy).

References

1. J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum, "YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages," in *Proceedings of the 20th International Conference Companion on World Wide Web*. ACM, 2011, pp. 229–232.
2. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A Nucleus for a Web of Open Data," *The Semantic Web*, pp. 722–735, 2007.

3. B. Vatant and M. Wick, "Geonames Ontology," *GeoNames, Accessed*, vol. 6, 2009.
4. H. Packer, S. Samangoeei, J. Hare, N. Gibbins, and P. Lewis, "Event Detection using Twitter and Structured Semantic Query Expansion," in *CIKM2012 - The First International Workshop on Multimodal Crowd Sensing*, 2012.
5. H. Packer, A. Smith, and P. Lewis, "MemoryBook: Generating Narrative from Lifelogs," in *Hypertext2012 - The Second International Workshop on Narrative and Hypertext Systems*, 2012.
6. E. Moxley, J. Kleban, and B. Manjunath, "Spirittagger: a Geo-Aware Tag Suggestion Tool Mined from Flickr," in *Proceeding of the 1st ACM international Conference on Multimedia Information Retrieval*, 2008, pp. 24–30.
7. D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *IEEE International Conference on Computer Vision*, vol. 2, p. 1150, 1999.
8. E. Moxley, J. Kleban, J. Xu, and B. Manjunath, "Not all Tags are Created Equal: Learning Flickr Tag Semantics for Global Annotation," in *IEEE International Conference on Multimedia and Expo.*, 2009, pp. 1452–1455.
9. L. Kennedy and M. Naaman, "Generating Diverse and Representative Image Search Results for Landmarks," in *Proceeding of the 17th International Conference on World Wide Web*, 2008, pp. 297–306.
10. M. Maala, A. Delteil, and A. Azough, "A conversion process from flickr tags to rdf descriptions," in *BIS 2007 Workshops*, 2008, p. 53.
11. H. Kawakubo and K. Yanai, "GeoVisualRank: a Ranking Method of Geotagged Images Considering Visual Similarity and Geo-Location Proximity," in *Proceedings of the 20th International Conference Companion on World Wide Web*, 2011.
12. B. Sigurbjörnsson and R. van Zwol, "Flickr Tag Recommendation Based on Collective Knowledge," in *Proceeding of the 17th International Conference on World Wide Web*, 2008, pp. 327–336.
13. J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua, "Inferring Semantic Concepts from Community-Contributed Images and Noisy Tags," in *Proceedings of the 17th ACM International Conference on Multimedia*, 2009, pp. 223–232.
14. J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *ICCV*, October 2003, pp. 1470–1477.
15. J. S. Hare, S. Samangoeei, and D. P. Dupplaw, "OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images," in *Proceedings of ACM Multimedia 2011*, ser. MM '11. ACM, 2011, pp. 691–694.
16. J. Hare, S. Samangoeei, D. Dupplaw, and P. Lewis, "Imageterrier: An extensible platform for scalable high-performance image retrieval." in *The ACM International Conference on Multimedia Retrieval (ICMR 2012)*, 2012.
17. D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, January 2004.
18. J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in *CVPR*, 2007.
19. M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, 2008.
20. H. Jegou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," in *Proceedings of the 10th European Conference on Computer Vision*, 2008, pp. 304–317.
21. D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, pp. 2161–2168.
22. Y. Avrithis, G. Toliás, and Y. Kalantidis, "Feature map hashing: Sub-linear indexing of appearance and global geometry," in *in Proceedings of ACM Multimedia*, Firenze, Italy, October 2010.