Summary Generation for Toponym-Referenced Images using Object Type Language Models

Ahmet Aker & Robert Gaizauskas
Department of Computer Science
University of Sheffield
Sheffield, S1 4DP, UK
A.Aker, R.Gaizauskas@dcs.shef.ac.uk

Abstract

This paper presents a novel approach to automatic captioning of toponym-referenced images. The automatic captioning procedure works by summarizing multiple web-documents that contain information related to an image's location. Our summarizer can generate both query-based and language model-biased multi-document summaries. The models are created from large numbers of existing articles pertaining to places of the same "object type". Evaluation relative to human written captions shows that when language models are used to bias the summarizer the summaries score more highly than the non-biased ones.

Keywords

Multi-Document Summarization, Image Captioning, Language Models, Statistical Methods, NLP

1 Introduction

In recent years the number of images on the web has grown immensely, facilitated by the development of cheap digital hardware and the availability of online image sharing social sites. Many of these images are tagged only with place names or contain minimal captions that include locational information. This small amount of textual information associated with the image is of limited usefulness for image indexing, organization and search. What would be useful is a means to generate or augment captions automatically based on existing data.

Attempts towards automatic generation of image captions have been previously reported. Deschacht & Moens [6] and Mori et al. [14] generate image captions automatically by analyzing image-related text from the immediate context of the image, e.g. the surrounding text in HTML documents. The authors identify named entities and other noun phrases in the image-related text and assign these to the image as captions. Other approaches create image captions by taking into consideration image features (colour, shape and texture) as well as image-related text [22, 14, 4, 7, 3, 15, 8]. These approaches analyze only the immediate textual context of the image. However, generating image captions based on the immediate context of the image can result in an image description which does not describe the image at all. Marsch & White [13] argue that the content of an image and its immediate text have little semantic agreement and this can, according to Purves et al. [16], be misleading to image retrieval. Furthermore, these approaches assume that the image has been obtained from a document. In cases where there is no document associated with the image, which is the scenario we are principally concerned with, these techniques are not applicable.

In this paper, we propose a technique for automatic image captioning or caption enhancement starting with only a set of place names pertaining to an image. The technique applies just to images of static features of the built or natural landscape (e.g. buildings, mountains, etc.) and not to images of objects which move about in such landscapes (e.g. people, cars, clouds, etc.).

Our approach is based on extractive multi-document summarization techniques, where the documents to be summarized are web-documents retrieved using the place names associated with an image. In earlier work [1] we have shown that in this scenario query-based summaries outperform generic summaries, i.e. extractive summaries of multiple web pages retrieved using the place names which bias the summarizer to include sentences mentioning these place names tend to be better than generic summaries of the same pages. However, the resulting summaries were still far from ideal. We examined information selected by humans for inclusion in a caption from the same place-name-retrieved web-documents made available to the summarizer and observed high levels of agreement between humans on which information to include. This led us to hypothesize that humans have a conceptual model of what is salient regarding a certain scene or object type (e.g. church, bridge, etc.) and that they use this in providing a description of the scene or object. Our qualitative analysis of Wikipedia articles (section 2) confirmed this hypothesis.

Given the observation that humans appear to have a conceptual model of what is salient regarding a specific object type, the question arises as to whether we can represent or approximate such a conceptual model in a way that allows us to improve content selection for our caption summaries. While there are many ways this could be done, one simple way is to view a corpus of descriptions of objects of a given type as containing an implicit model of that type and use language models derived from the corpus to bias sentence selection by an extractive summarizer.

In this paper we explore the use of signature words [12] and language models [21] to represent such conceptual models and investigate their impact on the quality of automatically generated image captions. Our results show that using these conceptual models does indeed improve the results over those of a standard query-based summarizer. In the following we first describe how the object type corpora were collected (section 2) and how language models are generated from these corpora (section 3). Next, we describe the set of our images, their categorization by object type and the retrieval of related web-documents (section 4). In section 5 we present the multi-document summarizer used to caption images. We discuss the results of evaluating automatic summaries against the human created captions in section 6, and conclude the paper in section 7.

2 Object Type Corpora

An object type corpus for our purposes is a collection of texts about a specific static object type such as *church, bridge, etc.* Objects can be named places or locations such as *Parc Guell, etc.* To refer to such object names we use the term *toponym*.

To build object type corpora we categorized Wikipedia articles about places by object types. For this categorization a Wikipedia dump[1] was used. The object types were identified automatically using *Is-A* patterns in the fashion of [10] and as described in [9]. The Is-A patterns were applied to the first ten sentences of each article. They match sentences which contain the type description of an object such as *... is a ...<object type>*. For *Westminster Abbey*, for instance, our Is-A patterns found the sentence which contains *... is a ... church*, extracted *church* as an object type from this sentence and assigned the article about the abbey to the *church* category. In this way we collected 107 categories containing articles about places around the world (cf. Table 1).

To assess the accuracy of the categorization we randomly selected 35 object type corpora and 50 articles from each corpus. Then we checked for each of these articles whether it is correctly assigned to its object type. Finally, we calculate an accuracy value for each object type by dividing the number of correctly assigned articles by 50 (cf. Table 2). We observed an average accuracy of 80% for all 35 object types.

We examined articles about different objects of the same type to investigate whether they contained recurring information. For this analysis we randomly selected 15 different object types from our entire set of 107. From each object type corpus we selected 20 articles about different objects. For each of the 15 object types we read all 20 associated articles and manually identified information that was repeated in at least two of the 20 articles. For illustration Table 3 shows the results of the analysis for three object types. From Table 3 we can observe that for each object type there is a common case of information used to describe instances of that type. This supports our hypothesis that humans have a shared idea about what is important information for an object type. Capturing this shared idea in conceptual models about object types could be used to bias a summarizer towards sentences that contain the information contained in the models.

Table 1: *Object types and the number of articles. Object types which are bold are covered by our image set.*

village 39970, school 15794, city 14233, organization 9393, **university** 7101, **area** 6934, **district** 6565, airport 6493, **island** 6400, **railway station** 5905, **river** 5851, company 5734, **mountain** 5290, **park** 3754, **college** 3749, **stadium** 3665, **lake** 3649, **road** 3421, country 3186, **church** 3005, way 2508, **museum** 2320, **railway** 2093, **house** 2018, arena 1829, field 1731, club 1708, shopping centre 1509, highway 1464, **bridge** 1383, **street** 1352, **theatre** 1330, bank 1310, property 1261, **hill** 1072, **castle** 1022, forest 995, court 949, hospital 937, peak 906, bay 899, **skyscraper** 843, **valley** 763, **hotel** 741, **garden** 739, **building** 722, market 712, **monument** 679, port 651, sea 645, **temple** 625, **beach** 614, **square** 605, store 547, campus 525, **palace** 516, **tower** 496, cemetery 457, **volcano** 426, **cathedral** 402, **glacier** 392, **residence** 371, dam 363, **waterfall** 355, **gallery** 349, **prison** 348, **cave** 341, **canal** 332, restaurant 329, path 312, observatory 303, **zoo** 302, coast 298, **statue** 283, **venue** 269, **parliament** 258, shrine 256, desert 248, synagogue 236, bar 229, **ski resort** 227, arch 223, landscape 220, **avenue** 202, casino 179, farm 179, seaside 173, waterway 167, tunnel 167, ruin 166, **chapel** 165, **observation wheel** 158, **basilica** 157, woodland 154, wetland 151, cinema 144, **gate** 142, **aquarium** 136, entrance 136, opera **house** 134, **spa** 125, shop 124, **abbey** 108, **boulevard** 108, **pub** 92, bookstore 76, **mosque** 56

Table 2: *Object types and the categorization accuracy.*

Object Type	Accuracy	Object Type	Accuracy
shopping center	0.9	**ski resort**	1.0
mountain	0.92	highway	0.82
railway station	1.0	mosque	0.66
waterfall	0.88	street	0.58
landscape	0.5	restaurant	0.86
island	0.92	**airport**	1.0
area	0.64	volcano	0.92
village	0.96	zoo	0.96
arena	0.96	wetland	0.79
bank	0.74	monument	0.62
university	0.98	building	0.52
park	0.96	gallery	0.725
museum	0.7	canal	0.82
temple	0.74	tower	0.52
prison	0.83	residence	0.8
aquarium	0.62	castle	0.86
bridge	0.72	waterway	0.83
river	0.94	**average accuracy**	**0.80**

3 Constructing Models

For constructing primitive conceptual models of shared information about object types we use two approaches: signature words [12] and generative language models as commonly used in information retrieval [21]. Using these two approaches we build uni-gram and bi-gram models for each object type using the corpus for that type constructed from Wikipedia articles as described above.

3.1 Signature Words

Signature words are a family of related terms [12]. Lin and Hovy use these terms to bias the sentence selection during the summarization process when creating topic-oriented summaries. They classify documents from the TREC collection as relevant or non-relevant for each given topic. Then, based on the relevant and non-relevant documents they generate for each topic a set of topic related terms or signature words. For each term in the set a weight is generated which expresses the importance of the term to the topic. The non-relevant documents are used to filter non-specific words from the topic-related documents. In the summarization process each sentence from the documents to be summarized is checked for whether it contains any word from the set of signature words. The score of the sentence is the sum of the weights of signature words it contains. Lin and Hovy showed that signature words lead to better summaries. There-

[1] English Wikipedia dump from 24/07/2008

Table 3: *Information commonly provided among the 20 Wikipedia articles for each object type.*

river:	where it originates; where it flows and ends/empties; length; other water bodies it joins; size of the area it drains; how fast it flows; tributaries it has;amount of water it discharges annually on average; location
church:	architecture; size (height, width); type of church (catholic, etc.); foundation year; architect; location;
mountain:	location; height(above see level); range; structure/shape; comparison to other mountains; when it was first climbed

fore we investigated the usefulness of this idea for the automatic image captioning task.

Similarly to Lin and Hovy we use our object type corpus to generate signature words. For each object type corpus we generate a uni-gram and a bi-gram signature word model:

$$ngram = \{corpus, [(ngram_1, score_1), .., (ngram_n, score_n)]\} \quad (1)$$

where *ngram* is either a single word (uni-gram) or two words (bi-gram). Lemmas of the words are used for both uni-gram and bi-gram models[2]. The score we use is the count of the n-gram lemma over the entire corpus divided by the most frequently occurring n-gram (to ensure that the n-gram score ranges between 0 and 1).

3.2 Language Models

Language models are used in different fields with different purposes. In information retrieval (IR), for instance, language models are used to retrieve documents relevant to a query. For each document a distinct n-gram language model is derived and used to estimate the probabilities of producing each term in the query [21]. The query is treated as a generation process, i.e. based on each language model the probability of generating each term in the query is computed. The probability of generating the query is the product of terms occurring in the query. Finally, the documents are ranked in descending order based on the probability assigned to the query. Therefore, if terms of a document lead to higher generation probabilities, the more relevant this document is to the query.

As an alternative to the signature word method we also generated language models from the object type corpora. Similar to [21] our language models are used in a generative way, i.e. we calculate the probability that a sentence is generated based on an n-gram language model. As for the signature word models we generate a uni-gram and a bi-gram model from each object type corpus:

$$ngram = \{corpus, [(ngram_1, prob_1), .., (ngram_n, prob_n)]\} \quad (2)$$

where again *ngram* is either the lemma of an uni-gram or bi-gram. $prob_i$ is the probability of an n-gram calculated using Good-Turing estimation:

$$prob(ngram) = \frac{(r+1)\frac{E(N_{r+1})}{E(N_r)}}{N} \quad (3)$$

where r is the number of times an n-gram is seen, N_r is the number of different n-grams seen exactly r times in the entire corpus, $E(N_r)$ is the expected value of N_r and N is the number of words in the entire corpus. However, in case $r=0$ (an n-gram is not seen) the probability is calculated as $E(N_1)/E(N_0N)$. N_0 is the number of n-grams which have not been seen. It is calculated by taking the square of the number of all seen n-gram types minus their sum.

4 Images & related Documents

Our image collection has 203 different images which are toponym-referenced, i.e. are assigned toponyms. The subjects of our images are locations around the world such as *Parc Guell, Edinburgh Castle*, etc. We manually categorized these images by object type. For each image we used its toponyms to search for a Wikipedia article using the Yahoo! search engine. We then selected the object type of the image from the Wikipedia article. For the image showing *Westminster Abbey*, for instance, we used the toponym *Westminster Abbey* to retrieve the Wikipedia article about the abbey, selected from this article the object type *church* and assigned the image showing the abbey to the object type category *church*. This process was repeated for our entire image set. Our images cover 60 of the 107 object types (cf. Table 1).

To generate automatic captions for these images we automatically retrieved the top ten related web-documents for each image from the Yahoo! search engine using the toponym associated with the image as a query. The text from these documents was extracted using an HTML parser and passed to the summarizer.

5 Summary Generation

The image captions are generated using *the-MDS* (the-multi-document summarizer), an extractive, language independent, multi-document, query-based summarization system implemented in Java. It uses a single cluster approach to summarize n related documents which are given as input. The summarizer creates image captions in a three step process. First, it applies shallow text analysis to the given documents. Then extracts features from the document sentences. Finally, it performs sentence selection to create the summary. The latter two tasks are language independent and can be performed for any UTF-8 encoded language. This means that *the-MDS* needs only a shallow text analyzer for any specific language in order to perform summarization. The three steps are described in more detail in the following subsections.

5.1 Shallow Text Analysis

The-MDS first applies shallow text analysis including sentence detection, tokenization, lemmatization and POS-tagging to the given documents using the OpenNLP tools.

5.2 Feature Extraction

After text analysis, *the-MDS* represents each sentence in the documents as a vector, where each vector position contains a term (word) and a value which is a product of the *term frequency* in the document and the *inverse document frequency (IDF)*, a measurement of the term's distribution over the set of documents [18]. The IDF table is generated from the n related documents. Furthermore, *the-MDS* enhances the sentence vector representation with four further features:

1. ***querySimilarity***: Sentence similarity to the query.
2. ***sentencePosition***: Position of the sentence within its document. The first sentence in the document gets the score 1 and

[2] Lemmatizing was performed using OpenNLP tools, http://opennlp.sourceforge.net/.

the last one gets $\frac{1}{n}$ where n is the number of sentences in the document.

3. ***centroidSimilarity***: Similarity to the centroid.
4. ***starterSimilarity***: A sentence gets a binary score if it starts with the query term (e.g. *Westminster Abbey*, *The Westminster Abbey*, *The Westminster* or *The Abbey*) or with the object type, e.g. *The church.*

For calculating vector similarities (*querySimilarity* and *centroidSimilarity*), the cosine similarity measure is used [19]. If there is an object type model, then for each sentence in the documents an additional fifth feature, the similarity to the given model (*modelSimilarity*), is added. In case of signature words this *modelSimilarity* is the sum of scores (*score*) of n-grams from a sentence S found also in the signature word model M (cf. Formula 4).

$$modelScore(S, M) = \sum_{ngram \in M \cap S} score_{ngram} \tag{4}$$

The *modelSimilarity* score with language models is calculated according to Formula 5.

$$modelScore(S, M) = \prod_{ngram \in s} (prob_{ngram} + 1) \tag{5}$$

In this case the *modelSimilarity* score of a sentence S is the product of scores (*prob*) of its n-grams where the *prob* values are obtained from the language model *M*. Finally, the feature vector representation of each sentence is passed to the sentence scoring process.

5.2.1 Sentence Scoring

We have two different approaches (signature word and language models) to determine the value for the *modelSimilarity* score. Both models, however, produce different value ranges for the same feature. To unify this score we apply a technique similar to the one described by Alfonseca et al. [2]. The authors produce a final ranked list for sentences from three different ranked lists for the same sentence by positioning the sentence which occurs in the top position in all three lists also in the top position of the final ranked list.

Following this idea *The-MDS* calculate the final sentence score. First, the first four features are used in a weighted linear combination to rank the sentences based on Formula 6.

$$S_{firstScore} = \sum_{i=1}^{n} feature_i * weight_i \tag{6}$$

The values for the weights are set to *.3* for the *querySimilarity*, *.1* for the *sentencePosition*, *.8* for the *centroidSimilarity* and *.9* for the *starterSimilarity*. We obtained these values empirically based on a set of 20 images selected randomly from our larger corpus of images. None of these 20 images is contained in the image set that we use for our evaluation. For this set of 20 images we generate summaries with different weight-value combinations, compare these summaries with human written captions and keep the weight-value combination which produces a summary with the highest ROUGE score.

The first ranking produces a ranked list of sentences in descending order by the $S_{firstScore}$. Then *the-MDS* uses the *modelSimilarity* feature to produce a second ranked list. Like the first ranked list the second list contains in its first position the sentence with the highest score. Finally, *the-MDS* combines these two lists to a final ranked list which is used to generate the summary. To produce the final list *the-MDS* takes for each sentence its position from the first and second ranked list and adds this sentence to the final list with a final score which is calculated using Formula 7.

$$S_{finalScore} = pos_{firstList} + 0.1 * pos_{secondList} \tag{7}$$

5.3 Sentence Selection

After the scoring process, *the-MDS* selects sentences for summary generation by selecting the sentence from the first position from the final list, followed by the next sentence in the list until the compression rate is reached. As in [17], before a sentence is selected a similarity metric for redundancy detection is applied to each sentence to decide whether a sentence is distinct enough from already selected sentences to be included in the summary or not. *The-MDS* measures lemma overlap between the words of the current sentence with the lemmas of previous selected sentences and includes the current sentence to the summary if the similarity measure is less than 30% which is obtained experimentally based on our training set images.

Using *the-MDS*, query-based (using first four features) and model-biased (using all five features) summaries are generated for the image-related documents obtained from the web. Each summary contains a maximum of 200 words. The queries used are the toponyms.

6 Evaluation

To evaluate our approach we compared the automatically generated summaries against model captions written by humans. Model captions were generated based on image captions taken from *Virtualtourist*[3]. Virtualtourist is one of the largest online travel communities in the world containing 3 million photos with captions (in English) of more than 58,000 destinations worldwide.

As with all information found in online knowledge sharing systems, there is no quality check for Virtualtourist captions. Members can describe places in anyway they want, resulting in image captions of different length, coherence, focus, grammaticality etc. To ensure a good standard for our model captions we asked 11 human subjects to generate up to four model captions per object by modifying Virtualtourist captions. The modifications included deleting personal information, ensuring consistency and coherence of the text and generating a summary of 190-210 words in length (because our automatic summaries have similar word counts). An example model summary about *Parc Guell* is shown in Table 6. For comparison between summaries the ROUGE metric [11] is used. ROUGE compares automatically generated summaries against human-created reference summaries and can be used to estimate content coverage in an automatically generated summary. Following the Document Understanding Conference (DUC) [5] evaluation standards we use ROUGE 2 and ROUGE SU4 as evaluation metrics. ROUGE 2 gives recall scores for bi-gram overlap between the automatically generated summaries and the reference ones. ROUGE SU4 allows bi-grams to be composed of non-contiguous words, with a maximum of four words between the bi-grams.

[3] www.virtualtourist.com

Table 4: *ROUGE scores for the first document (*F*), Wikipedia (*W*) and the query-based (qB) baselines. The last 3 columns show* z *scores and the significance of the Wilcoxon signed ranked test.*

Recall	F	W	qB	F<W	F<qB	W>qB
R2	.045	**.095**	.066	-10.4***	-7***	-8.9***
RSU4	.081	**.14**	.114	-10.8***	–8.6***	-8.6***

Table 5: *ROUGE results for uni-gram and bi-gram biased models (signature words (WS) and language models (WL)). The first 2 rows show the results for uni-gram and the last 2 rows for the bi-gram models. The last 4 columns show* z *scores and the significance of the Wilcoxon signed ranked test.*

Recall	WS	WL	WS<WL	WL>qB	WS>qB	WL<W
R2	.068	**.07**	-1.9	-4***	-1.5	-8.3***
RSU4	.115	**.118**	-2.6**	-4.8***	-1.5	-7.3***
R2	.068	**.071**	-2.4*	-5.2***	-1.9	-8***
RSU4	.115	**.119**	-4***	-5.9***	-.67	-7.3***

As baselines for evaluation we use three summary types. Firstly, we generate summaries for each image using the top-ranked non Wikipedia document retrieved in the Yahoo! search results for the given toponyms. From this document we create a baseline summary by selecting sentences from the beginning until the summary reaches a length of 200 words. As a second baseline we use the Wikipedia article for a given toponym list from which we again select sentences from the beginning until the summary length limit is reached. Thirdly, we include query-based summaries generated without language models. Table 4 shows the ROUGE scores when baseline summaries are compared to the Virtualtourist model summaries. To assess the statistical significance of ROUGE score differences between multiple summarization results we performed a pairwise Wilcoxon signed-rank test with Bonferroni correction[4] for multiple testing.

Both Wikipedia baseline and query-based summaries score significantly higher than the first document baseline. The Wikipedia baseline scores are also significantly higher than the query-based ones. It follows from these results that the Wikipedia baseline summaries have the best coverage of the content in our model captions. Table 6 shows the Wikipedia baseline summary about *Parc Guell*.

Using the same Virtualtourist model captions we also evaluated the uni-gram and bi-gram model-biased summaries. It should be noted that the set of documents we used to generate our summaries do not contain any Virtualtourist related sites, as these are used to generate our model summaries. The results are given in Table 5 and show that the highest scoring summaries are the ones biased with language models. Table 6 shows the language model-biased summary about *Parc Guell*. In both uni-gram and bi-gram models the language models score significantly higher than signature word models as well as query-based summaries. The signature words summaries perform moderately higher than query-based summaries. However, both signature words and language model summaries are significantly lower than the Wikipedia baseline summaries (Due to limited space Table 5 shows only the comparison between the language model and Wikipedia baseline summaries). These results show that language model biased summaries lead to significant improvement in ROUGE results compared to the query-based summaries. One reason for this might be that the query-based summarizer takes relevant sentences according to the query given to it and does not take into more general consideration the information typically provided for the, albeit simple, object type. Our language models are one way of capturing shared interests about some particular object type. To assess whether and to what extent language model biased summaries contain more shared information than query-based ones, we also qualitatively analyze the sentences in query-based and language model-biased summaries. First, we delete all sentences that occur in both summary types to focus only on differences between the two methods. Then, for each remaining sentence, we check whether it carries one of the facets of information about an object type commonly presented in Wikipedia articles (cf. section 2). If this is the case, the sentence is selected. Finally, we count the number of selected sentences in query-based and language model-biased summaries. Language model-biased summaries covered 76 sentences containing shared information whereas query-based summaries covered only 34 such sentences. While this is not the total number of sentences containing shared information, it highlights the differences between the two summarization methods with respect to capturing shared information about object types. Language model-biased summaries contain 51% more of the information commonly provided in the Wikipedia articles than the query-based summaries. This implies that the model-biased summaries do indeed help to bias the summarizer towards information commonly used for certain object types, which in turn improves the quality of summaries or image captions.

6.1 Discussion

There are several application areas for our automatically generated image captions. They could provide useful information about objects to interested users, e.g. a tourist who is looking for some basic information about a place to visit. Also they could be used as a way to automatically index images. The automatic summary shown in Table 6 could serve both these purposes. It contains only sentences relevant to *Parc Guell* without any unrelated information. Furthermore, the summary contains terms such as *park*, *Barcelona centre*, *Gaudi's creations*, etc. These terms could be used to index an image showing *Parc Guell*, which would potentially provide better indexing than using the park's name only. Sanderson & Kohler [20], for example, analyzed search engine queries containing place names and other geographic terms such as object types (street, island, lake, etc.), address and direction information, etc. They showed that more than 40% of the queries contained other geographic terms beside the place name. Thus indexing images with the place name and the terms from the automatically

[4] After Bonferroni correction all effects are reported at a *p=.0167* level of significance. We use the following conventions for indicating significance level in the tables: *** = p < .0001, ** = p < .001, * = p < .0167 and no star indicates non-significance. We also use Wilcoxon test for all pairwise comparisons reported in the text, in which case no correction is applied, and the results are reported relative to significance level *p<.05*.

Table 6: *Model, Wikipedia baseline and language model-biased summary for Parc Guell.*

Model Summary	Wikipedia baseline summary	Language model-biased summary
One of the star attractions of Barcelona is the Parc Guell, the second most visited park after the Ciutadella probably only because of its location in the north of the city. The park owes its magnetic attraction to the fact that it was designed by Gaudi. In creating the park, Gaudi used shapes which harmonized with the landscape. Always aware of the struggle betwen man and nature, he built a complex garden of staircases, animal like sculptures, curvy ramps, and viaducts. Today, the park is declared a monument of world interest by UNESCO and has had this status since 1984. The most interesting part of the park consists of the large snake bench. Each part of the bench has a different looking pattern and color. Then down below this part is the famous lizard. Above this more popular area are pathways to walk on and even these are decorated with palm trees and unique benches made of rock. The best way to get there is with the bus. The bus number 24 bus stops close to the park. The bus stop is located near placa catalunya, The Metro is a much longer walk away from the park. Admission to the park is free!	The park was originally part of a commercially unsuccessful housing site, the idea of Count Eusebi Guell, whom the park was named after. It was inspired by the English garden city movement; hence the original English name Park (in the Catalan language spoken in Catalonia where Barcelona is located, the word for "Park" is "Parc", and the name of the place is "Parc Guell" in its origin language). The site was a rocky hill with little vegetation and few trees, called Montana Pelada (Bare Mountain). It already included a large country house called Larrard House or Muntaner de Dalt House, and was next to a neighborhood of upper class houses called La Salud. The intention was to exploit the fresh air (well away from smoky factories) and beautiful views from the site, with sixty triangular lots being provided for luxury houses. Count Eusebi Guell added to the prestige of the development by moving in 1906 to live in Larrard House. Ultimately, only two houses were built, neither designed by Gaudi.	The park is huge with many different sights to enjoy. The park is full of narrow twisting pathways which meander through thepark. The park contains amazing stone structures (see below), stunning tiling and fascinating buildings. In every sense, Barcelona has become one of the hottest destinations in Spain. The park was built between 1900-1914, originally planned to be a garden city on the estate of Eusebi Guell. More: everything looks like it wasn't created by man, but by nature. Great park, quite a lot so see as you walk around, some mad buildings! The walk from the metro will take you about 20 mins. Park Gueell is another of Gaudi's creations and lies north of Barcelona centre, 20 mins walk from Lesseps Metro (Green Line, L3). The extraordinary craftsmanship and unusual use of materials and plants throughout the park catch and delight the eye, making Parc Guell one of the great parks of the world. Ultimately, only two houses were built, neither designed by Gaudi.

generated caption or summary could indeed lead to better retrieval. This would be the case for all search engine queries which do not contain a specific place name but rather are more general query such as *parks in Barcelona*. However, one could argue that the same benefits would be achieved by simply taking Wikipedia articles as image captions, rendering multi-document summarization unnecessary for captioning. Our results showed that initial sentences from Wikipedia articles are indeed a tough baseline for evaluation of image captions. One problem with this, however, is that Wikipedia does not contain an article for every location that may be described on the web. In our larger image set, for instance, no Wikipedia article exits for 30 images. This gives us the motivation to further develop multi-document summarization techniques for image captioning.

7 Conclusion

In this work we have proposed an approach to automatic captioning of toponym-referenced images using query-based multi-document summarization techniques. We showed that query-based summarizers biased with a language model for a specific object type perform significantly better than standard query-based summarizers without such models. The language models are generated from object/scene type corpora built from Wikipedia articles which have been automatically categorized by object type. In future work we plan to investigate alternative ways of modelling conceptual knowledge about object types and also ways of producing more coherent summaries. We also plan to investigate the application of the same technique to other languages.

8 Acknowlegment

The research reported was funded by the TRIPOD project supported by the European Commission under the contract No. 045335. We would like to thank Mark Greenwood for discussions and comments.

References

[1] A. Aker and R. Gaizauskas. Evaluating automatically generated user-focused multi-document summaries for geo-referenced images. *Proc. of COLING08, Workshop on Multi-source, Multilingual Information Extraction and Summarization (MMIES2)*, 2008.

[2] E. Alfonseca, M. Okumura, J. Guirao, and A. Moreno-Sandoval. Googling answers models in question-focused summarisation. *Document Understanding Conference*, 2006.

[3] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3:1107–1135, 2003.

[4] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *International Conference on Computer Vision*, volume 2, pages 408–415. Vancouver: IEEE, 2001.

[5] H. Dang. Overview of DUC 2006. *National Institute of Standards and Technology*, 2006.

[6] K. Deschacht and M. Moens. Text Analysis for Automatic Image Annotation. *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007.

[7] P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. *In 7th European Conference on Computer Vision (ECCV)*, 4:97–112, 2002.

[8] Y. Feng and M. Lapata. Automatic Image Annotation Using Auxiliary Text Information. *Proc. of Association for Computational Linguistics*, 2008.

[9] T. Gornostay and A. Aker. Development and Implementation of Multilingual Object Type Toponym-Referenced Text Corpora for Optimizing Automatic Image Description. *Proc. of the 15th International Conference on Computational Linguistics*, 2009.

[10] M. Hearst. Automatic acquisition of hyponyms from large text corpora. *Proc. of the 14th conference on Computational linguistics*, 1992.

[11] C. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. *Proc. of the Workshop on Text Summarization Branches Out*, 2004.

[12] C. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *Proc. of the COLING Conference*, 2000.

[13] E. Marsh and M. White. A taxonomy of relationships between images and text. *Journal of Documentation*, 2003.

[14] Y. Mori, H. Takahashi, and R. Oka. Automatic word assignment to images based on image division and vector quantization. *Proc. of RIAO 2000: Content-Based Multimedia Information Access*, 2000.

[15] J. Pan, H. Yang, P. Duygulu, and C. Faloutsos. Automatic image captioning. *Multimedia and Expo. ICME'04. IEEE International Conference on Multimedia and Expo*, 2004.

[16] R. Purves, A. Edwardes, and M. Sanderson. Describing the where–improving image annotation and search through geography. *1st Intl. Workshop on Metadata Mining for Image Understanding, Funchal, Madeira-Portugal*, 2008.

[17] H. Saggion. Topic-based Summarization at DUC 2005. *Document Understanding Conference*, 2005.

[18] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 1988.

[19] G. Salton and M. Lesk, E. Computer evaluation of indexing and text processing. *Journal of the ACM*, 1968.

[20] M. Sanderson and J. Kohler. Analyzing geographic queries. In *SIGIR Workshop on Geographic Information Retrieval*, 2004.

[21] F. Song and W. Croft. A general language model for information retrieval. *Proc. of the eighth international conference on Information and knowledge management*, 1999.

[22] T. Westerveld. Image retrieval: Content versus context. *Content-Based Multimedia Information Access, RIAO 2000 Conference*, 2000.