Generating Descriptive Multi-Document Summaries of Geo-Located Entities Using Entity Type Models

Ahmet Aker and Robert Gaizauskas
Department of Computer Science, University of Sheffield, 211 Portobello, S1 4DP, Sheffield, United Kingdom.
E-mail: {a.aker, R.Gaizauskas} @dcs.shef.ac.uk

In this article, we investigate the application of entity type models in extractive multi-document summarization using automatic caption generation for images of geo-located entities (e.g., *Westminster Abbey*) as an application scenario. Entity type models contain sets of patterns aiming to capture the ways geo-located entities are described in natural language. They are automatically derived from texts about geo-located entities of the same type (e.g., churches, lakes). We integrate entity type models into a multi-document summarizer and use them to address the 2 major tasks in extractive multi-document summarization: *sentence scoring* and *summary composition*. We experiment with 3 different representation methods for entity type models: *signature words*, *n-gram language models,* and *dependency patterns*. We evaluate the summarizer with integrated entity type models relative to (a) a summarizer using standard text-related features commonly used in text summarization and (b) the Wikipedia location descriptions. Our results show that entity type models significantly improve the quality of output summaries over that of summaries generated using standard summarization features and Wikipedia summaries. The representation of entity type models using dependency patterns is superior to the representations using signature words and n-gram language models.

Introduction

Automatic text summarization aims to represent the topics found in one or more input documents to the user in a condensed form and so to reduce the time that the user spends reading all the documents (Jones, 1999; Mani, 2001). Two different approaches to automatic document summarization have been developed: *extractive* and *abstractive*. In extractive document summarization, the most important sentences from the input document are taken as the condensed form of the document and presented to the user in the order that they occur in the original document until a stipulated summary length or compression ratio is reached. The compression ratio indicates the number of sentences or words, relative to the number of complete sentences or words in the text that the summary should contain. By contrast, abstractive approaches aim to rephrase the content identified as relevant in fewer words than the original text.

A summary can be generated from a *single document* or from *multiple documents*. In both cases, there is a distinction between *generic* and *query-focused* text summarization. In generic text summarization, the summary content is determined based only on the content of the input documents. In query-focused text summarization, the summarizer is given a natural language *query* as input, which is used by the summarizer to bias its sentence selection toward the pieces of information closely related to that query. The query can take any form. For instance, the query can be an open-ended question about a person, such as “Who is X?” as formulated in the Document Understanding Conferences (DUC),[1] with “X” being the name of a person (Nenkova & McKeown, 2011). To help extract facts about the person, one could consider using a person type model. Such a model might capture what facts are typically provided about a person and the ways that they are described in existing texts. When generating a summary about a specific person X, the person type model could then be used to bias the summarizer’s sentence selection. In addition, the person type model could be used to mark each sentence with the type of

Received July 23, 2013; revised November 12, 2013; accepted December 9, 2013

© 2014 ASIS&T • Published online 15 May 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.23211

[1] http://duc.nist.gov/

information it contains, such as date of birth. This would help the summarizer to compose the summary by selecting sentences with unique facts and thus reduce redundancy within the summary. Finally, the model could be used to order the sentences in the summary. For example, a sentence that contains information about the date of birth of a person could be marked by the model as preceding the sentence containing the date of death of that person. Applying such relationships between the sentences during the summary composition can lead to more coherent summaries and avoid the common problem of the summary reading like a heap of information without any meaningful connection between the sentences.

In this work, we use query-focused, multi-document text summarization to generate a summary for an entity expressed in the query. Instead of a person, we use a *geo-located entity* in the query. Geo-located entities are static features of the built or natural landscape, such as a building, bridge, mountain, or river. We create *geo-located entity type models* to bias the summarizer's sentence selection, but also for redundancy reduction and sentence ordering within the summary. Entity type models contain sets of patterns aiming to capture the ways that the geo-located entities are described in natural language. Our models are learned offline from existing texts about different entities of the same type, such as church, river, mountain, and so on. We refer to such textual descriptions as *entity type corpora*. We apply our summarizer to the task of generating captions for images pertaining to geo-located entities.

Note, given that conceptualization is a general feature of human thinking, that the idea of entity type modeling is not limited to geo-located entity description generation but also applies to other domains and genres. Therefore, our technique is suitable not only for image captioning but in any application context that requires summary descriptions of instances of entity classes, where the instance is to be characterized in terms of the features typically mentioned in describing members of a class.

The article is organized as follows. We first introduce our application scenario and discuss related work in image caption generation. Next, we outline our method. Following this, we describe the text descriptions used to derive the entity type models. We next describe the three entity type modeling strategies—signature words, language models, and dependency patterns—and explain how they are applied to the existing text descriptions. We later present how we use the entity type models in the summarization process, followed by a description of our summarization system. We then present our experimental settings and discuss the results of both automatic and manual evaluations. These evaluation results cover the case where an entity type model is generated for a single entity type such as *church*. We also group entity types based on their purpose and appereance (e.g., church, cathedral, basilica, temple) and create entity type models from groups of entity types. Finally, we present and discuss the results for entity type models built from groups of entity types.

Application Scenario

The number of images available electronically is growing exponentially with the rapid development of online photo-sharing services and increasing prevalence of digital cameras and camera phones. In addition, many legacy photographs and other images are stored or archived. Effective access to these images is only possible if they are searchable, which presupposes that images are indexed and easily identifiable. However, typically, only limited textual information is available with each image, usually in the form of a set of keywords assumed to describe an image. Alternatively, it could be that no textual information is provided but that the images are tagged only with geocoordinates and compass information. Such a small or nonexistent amount of textual information associated with an image is of limited usefulness for image indexing, organization, and search. What would be useful is a means to automatically generate or augment captions for images on the basis of minimal input information. The generated captions could then be used for indexing purposes.

Automatic image captioning is a challenging task because it is not straightforward to decide what to include in a caption. One can capture any kind of object in the universe with an image, so the content of an image can be virtually anything we can see (abstract objects made by photo-montage are left out of consideration.) However, most objects are multifaceted, and it is not clear which aspects of the image the caption should address. For example, if we take an image of the *Matterhorn* (Figure 1), one could say that the image shows "a mountain covered with snow" or "Matterhorn" if it is known that it is indeed the *Matterhorn*. Alternatively, an interpretative description could be given (e.g., the image shows challenge, difficulty, etc.). To make the interpretation, the person writing the description needs more knowledge about the Matterhorn. Therefore, such

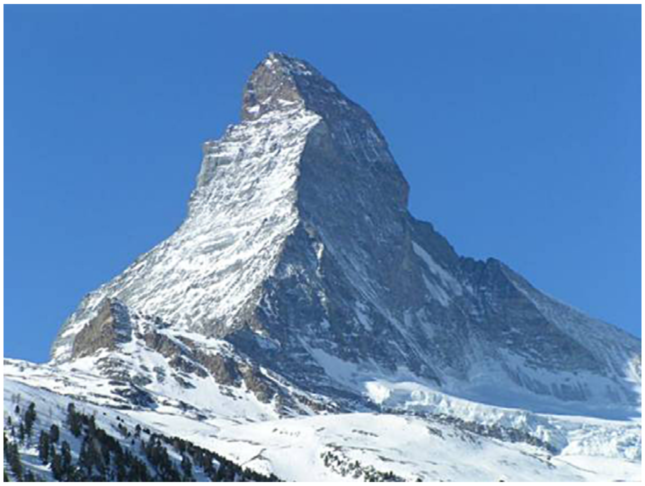

FIG. 1. Mountain of Matterhorn. The image is taken from Wikipedia. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

descriptions can vary from individual to individual and depending on use of the description because each individual will have different knowledge about the object(s) shown in the image and a different interpretation of this knowledge. In addition, the task or context of use also will have impact on the resulting descriptions.

To gain insight into what types of descriptions humans associate with images, a substantial number of investigations into how to classify images or image content in categories have been carried out. As a result, many classification schemas or analyses dedicated to categorization of image-related information have been proposed (e.g., Armitage & Enser, 1997; Eakins, 1998; Jorgensen, 1998; Jaimes & Chang, 2000). However, the classification schema proposed by Panofsky (1972) and modified by Shatford (1986) shown in the matrix in Table 1 has been used as the direct or indirect basis for all these works in the field of image classification.

Attempts toward automatic generation of image captions, which mostly address the *what* and *who* facets of the Panofsky–Shatford matrix have been reported. They can be divided into text-based and content-based image-caption-generation approaches. Text-based approaches generate image captions solely using texts related to the image and do not take image features such as color, texture, position, and so on into consideration. The content-based methods take image-related texts as well as the image features as input and output image captions based on these two input resources.

To our knowledge, the work of Deschacht and Moens (2007) is the only text-based approach in image captioning. The authors automatically generated image captions using associated text such as existing image captions, video transcripts, or surrounding text in web pages. They tried to identify entities (names of persons and objects) shown in the image. To do this, they first detected persons and objects in the associated text by applying automatic named-entity recognition. Then, they ranked the identified entities (person and object names) by assigning them salience weights (the importance of an entity in a text based on word statistics) and visualness measures (how likely an entity will be present in the image). Entities above a threshold of these measures are taken as captions for the image. In contrast to Deschacht and Moens, several different content-based approaches (Barnard & Forsyth, 2001; Barnard et al., 2003; Duygulu, Barnard, de Freitas, & Forsyth, 2002; Farhadi, Endres, Hoiem, & Forsyth, 2009; Farhadi et al., 2010; Feng & Lapata, 2008, 2010a, 2010b, 2010c; Gupta & Mannem, 2012; Kulkarni et al., 2011; Kuznetsova, Ordonez, Berg, Berg, & Choi, 2012; Mori, Takahashi, & Oka, 2000; Pan, Yang, Duygulu, & Faloutsos, 2004; Yao, Yang, Lin, Lee, & Zhu, 2010) made use of the text resources related to the image as well as image features to describe the image content. Although there are differences in problem formulation and application, the common idea presented by these studies is (a) to relate words or greater units such as sentences from the immediate textual context of an image to features or attributes extracted from the image and (b) use the high corelating text units as description of the image content.

The main drawback of these approaches is that they rely on texts associated with images. However, the associated text may have little semantic agreement with the content of the image, which can result in captions which do not describe the image at all (Marsh & White, 2003). Using these "wrong" captions for indexing purposes, for instance, can be misleading to image retrieval (Purves, Edwardes, & Sanderson, 2008). More important, these approaches assume that there exists a text associated with an image. This could be the case if the image has been obtained from a document, for example, or has some existing description or caption describing its content. However, this need not be the case. Where there is no document associated with the image or if no immediate text that describes its content exists, these techniques are not applicable. Captioning images with little or no associated text information is precisely the scenario with which we are concerned in his work.

Method

For our application scenario, we use query-focused, extractive multi-document summarization to generate captions or summaries for geo-located entities. In our system, the documents to be summarized are web documents retrieved using the name of the geo-located entity shown in an image (e.g., Eiffel Tower) as a query. The resulting image caption has the form of a short description or summary of the place in the image, which distinguishes it from captions in the form of lists of keywords generated in much previous work (e.g., Barnard et al., 2003; Duygulu et al., 2002; Farhadi et al., 2009; Pan et al., 2004). Therefore, in the

TABLE 1. The Panofsky–Shatford mode/facet matrix (Shatford, 1986).

Facets/modes	Specific of	Generic of	About
WHO?	Individually named persons, animals, things	Kinds of persons, animals, things	Mythical beings (Generic/Specific), abstractions manifested, or symbolized by objects or beings
WHAT?	Individually named events	Actions, conditions	Emotions, abstractions manifested by actions, events
WHERE?	Individually named geographic location	Kind of place geographic or architectural	Places symbolized (Generic/Specific), abstractions manifested by locale
WHEN?	Linear times; dates or periods	Cyclical time; seasons, time of day	Emotions or abstractions symbolized by or manifested by time

remainder of the article, we use the terms *caption*, (image) *description*, and *summary* interchangeably.

Extractive multidocument summarization presents several challenges. First, it is necessary to distinguish between summary-relevant and summary-irrelevant sentences. This is referred to as *sentence scoring* or *sentence ranking*. Summary-relevant sentences are those which are candidates for inclusion in the final summary and thus should be ranked or scored by the summarizer more highly than should the summary-irrelevant sentences. Once the sentences are scored, there is the challenge of composing the final summary from these sentences so that (a) the summary is informative (i.e., contains the most relevant pieces of information without exceeding a predefined length), (b) does not contain redundant information, and (c) is fluent to read. Constructing such a summary from a subset of scored sentences will be referred to as *summary composition*.

Previous work has identified several text-based features which are commonly used in sentence scoring (for a review, see Lloret & Palomar, 2012). These features are "universal" text features, in the sense that they capture a topic and other general properties of a text independently of what a text is about, and what kind of issue the summary should address. They may work well in some domains or genres, but not in others. For example, the *sentence position* feature indroduced by Baxendale (1958) indicates the position of the sentence within its document so that, for example, the first sentence in the document gets the highest score, and the score decreases toward the end of the document. This feature has been found useful in the news genre. For news articles, the first sentences in the article are worth including in the summary because they usually summarize the entire article (Baxendale, 1958; Kupiec, Pedersen, & Chen, 1995; Teufel & Moens, 1997). However, Kim, Le, and Thoma (2007) noted that this feature was not useful for scoring sentences in biomedical research papers. What may be useful in every domain is to capture how people think about the entities, events, and general topics of this domain. This involves identifying the types of information people associate with the topics of the domain and scoring the sentences which address them more highly. Unlike the direct text features, this involves a level of abstraction beyond the text, as sentences need to be categorized according to the information types that they address. By doing so, however, the foci of interest within a domain can be captured and addressed in the summaries, which may improve their quality.

For this reason, in addition to using the features commonly used for sentence scoring in previous work, our multi-document summarizer biases the sentence scoring according to an entity type model. Using entity type models in sentence scoring is central to our approach. It derives from the fact that humans can categorize things that they see in their environment. Cognitive psychology has offered several theories and substantial empirical evidence for existence of categories or concepts and an explanation of what constitutes them (Eysenck & Keane, 2005). These theories agree that concepts are characterized by sets of attributes, although they differ in whether a set of attributes is necessary and sufficient to define a concept (defining-attributes theories) or whether the concepts are more fuzzy in their specification in terms of attributes (prototype theories) so that some instances are more representative of a concept than are others.

If humans use concepts to organize knowledge about the world, then they will have ways to describe these concepts in natural language. We argue that to build a good summary about a geo-located entity (e.g., Eiffel Tower, Westminster Abbey, etc.), we need to select sentences which address the attributes specific to the concept into which this entity can be categorized (e.g., tower, church, etc.). This can be achieved if the sentence selection is biased according to an entity type model.

We derive entity type models automatically from texts describing entities of the same type. The models contain sets of patterns aiming to capture the ways that the entities are described in natural language. We investigate whether entity type models can help our summarization system to perform better sentence scoring.

We also apply entity type models for summary composition to address redundancy and sentence ordering.

The common approach to avoiding redundancy is to use a text-similarity measure to block the addition of a further sentence to the summary if it is too similar to one that is already included (e.g., Saggion & Gaizauskas, 2004). The similarity is controlled by a similarity threshold. Any sentence whose similarity is above the threshold is not included in the summary. The similarity threshold is either manually set to an arbitary value (Barzilay, McKeown, & Elhadad, 1999; Lin & Hovy, 2002; Sauper & Barzilay, 2009) or learned automatically from the data (Aker, Cohn, & Gaizauskas, 2012).

We use entity type models represented as dependency patterns (discussed later) to address redundancy. Our dependency patterns express specific types of information. We group the patterns into groups expressing the same type of information and then, during sentence selection, ensure that sentences matching patterns from different groups are selected to guarantee broad, nonredundant coverage of information relevant for inclusion in the summary.

For producing a fluent summary, we adopt the idea of using predefined categories for sentence ordering that has been reported by related work (Bollegala, Okazaki, & Ishizuka, 2010; Liakata, Teufel, Siddharthan, & Batchelor, 2010; Liddy, 1991; Teufel, 2010; Teufel & Moens, 2002). Liakata et al. (2010), for instance, worked with scientific papers and used predefined, manually created categories such as Background, Hypothesis, Motivation, Goal, Object, Method, Model, Experiment, Observation, Result, and Conclusion into which to map the input sentences. In the summarization process, they proposed using the categories in the given order and take for each category the highest ranking sentence to include in the summary. We follow related work and use the groups expressing the same information type as our predefined categories. We put these categories in the order that they also occur in manually written image descriptions and use this order while generating the summary.

Entity Type Corpora

We derive our entity type models from entity type corpora described in Aker and Gaizauskas (2009). We defined an entity type corpus as a collection of texts about a specific static entity type such as church, bridge, and so on. Entities can be named locations such as "Eiffel Tower." To build such entity type corpora, we categorized Wikipedia articles about places by entity type (see Figure 2).

The entity type of each article was identified automatically by running Is-A patterns (e.g. enity IS A church) over the first five sentences of the article. The authors reported 91% accuracy for their categorization process. The most populated of the categories identified (in total, 107 containing articles about places around the world) are shown in Table 2.

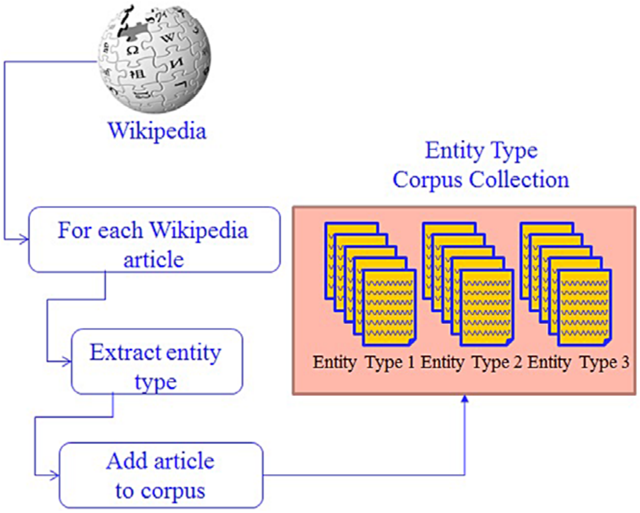

FIG. 2. Entity type collection procedure of Aker and Gaizauskas (2009). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Entity Type Models

We use three different methods for entity type modeling or deriving the entity type model feature from entity type corpora: *signature words, language modeling,* and *dependency patterns*. These methods differ in how they represent the collected entity type corpora as a model. We evaluate each method based on its impact on automatic image-description-generation performance and report the results. We report the results of the automatic evaluation using Recall-Oriented Understudy of Gisting Evaluation (ROUGE; Lin, 2004) and those of human readability assessment. The models are derived from descriptions belonging to a single entity type corpus such as church, but also from descriptions coming from groups of entity types such as *museum, opera house*, and *art gallery*.

Signature Words

Lin and Hovy (2000) introduced the notion of signature words for summarizing articles about news events, which they defined as a family of related terms. They used signature words to represent the topic in the input documents. The topic words are selected from the input documents by comparing them to preclassified texts on the same topic using the likelihood ratio λ (Dunning 1993), a statistical test to compute the likelihood of a word being a member of the set of relevant documents rather than the nonrelevant ones. For each word in the input documents, they computed the likelihood of the occurrence of that word in the preclassified topic text collection. Another likelihood value is computed using the same word and another text collection that is out-of-topic. If the word has higher likelihood for the topic text collection than for the out-of-topic one, then the word is taken as a signature for the topic; otherwise, the word is omitted from inclusion. They experimented with single signature words (unigrams), two consecutive words (bigrams), and three consecutive signature words (trigrams) and reported the best summary results using bigrams. In each case, they used lemmas of the words. As topics, the authors used overcrowded prisons, cigarette consumption, computer

TABLE 2. Entity types (80 urban, 27 rural) and the number of articles in each corpus.

Urban	Rural
School 15,794; city 14,233; organization 9,393; university 7,101; area 6,934; district 6,565; airport 6,493; railway station 5,905; company 5,734; park 3,754; college 3,749; stadium 3,665; road 3,421; country 3,186; church 3,005; way 2,508; museum 2,320; railway 2,093; house 2,018; arena 1,829; club 1,708; shopping center 1,509; highway 1,464; bridge 1,383; street 1,352; theatre 1,330; bank 1,310; property 1,261; castle 1,022; court 949; hospital 937; skyscraper 843; hotel 741; garden 739; building 722; market 712; monument 679; port 651; temple 625; square 605; store 547; campus 525; palace 516; tower 496; cemetery 457; cathedral 402; residence 371; gallery 349; prison 348; canal 332; restaurant 329; observatory 303; zoo 302; statue 283; venue 269; parliament 258; shrine 256; synagogue 236; bar 229; arch 223; avenue 202; casino 179; waterway 167; tunnel 167; ruin 166; chapel 165; observation wheel 158; basilica 157; 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	Village 39,970; island 6,400; river 5,851; mountain 5,290; lake 3,649; field 1,731; hill 1,072; forest 995; peak 906; bay 899; valley 763; sea 645; beach 614; volcano 426; glacier 392; dam 363; waterfall 355; cave 341; path 312; coast 298; desert 248; ski resort 227; landscape 220; farm 179; seaside 173; woodland 154; wetland 151

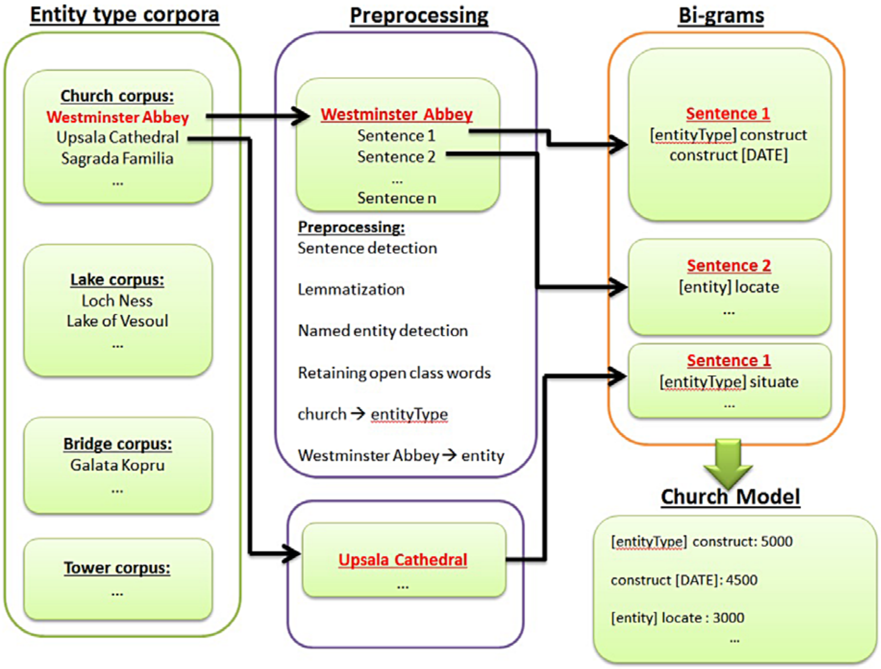

FIG. 3. Signature words (bigrams) model generation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

security, and solar power and the corresponding articles from the TIPSTER-SUMMAC (Tipster Text Summarization Evaluation) collection (Mani et al., 1999).

In the summarization process, each sentence from the input documents of a specific topic is checked for whether it contains any word from the set of signature words of that topic. The score of the sentence is the sum of the weights of signature words that it contains. Hovy and Lin (1998) integrated the signatures into the SUMMARIST (automated text) summarization system and compared the performance of signature words with two other features: sentence position[2] and term frequency-inverse document frequency (td-idf) (Manning et al., 2008). The authors reported that signature words outperformed the other two features, the worst performing feature being tf–idf.

We use signature words as one method for entity type modeling. We derive signature words from the Wikipedia articles describing geo-located entities of the same entity type.

Application to Entity Type Corpora

To derive a signature for each entity type corpus, we use the following formula and generate signature words containing unigram and bigram signatures:

$$\begin{aligned} \mathit{ngramModel} = (&\text{entityType}[(\text{ngram}_1, \text{freq}_1), \ldots, \\ &(\text{ngram}_n, \text{freq}_n)]), \end{aligned} \tag{1}$$

where ngram_x is either a single word (unigram) or two consecutive words (bigram). We do not use trigrams because, according to Lin and Hovy (2000), they are not a good choice for topic representation.

Figure 3 illustrates the signature model generation for the bigram case. Given an entity type corpus (e.g., a church corpus), we preprocess every Wikipedia article within the corpus by first performing sentence detection, lemmatization, and named entity detection. We also remove any word that is not an open-class word (noun, verb, adjective, or adverb). For these tasks, we use the OpenNLP tools.[3] We also replace any occurrence of a string denoting the entity type by the term "entityType" and the entity name by "entity" as shown in Figure 3. Finally, from each sentence we extract n-grams. Assume that Sentence 1 from the *Westminster Abbey* article in Figure 3 is:

The church was constructed in 1245.

After lemmatization, we obtain the sentence:

The church is construct in 1245.

After named entity detection, we obtain the following:

The church is construct in [DATE].

After filtering out all words that are not open-class, we obtain the sentence:

church construct [DATE]

We replace church by [entityType] and get:

[entityType] construct [DATE].

Finally, from this sentence, we generate n-grams. Figure 3 shows the case for bigrams. Since the n-grams also will occur in several sentences of other Wikipedia articles within the entity type corpus, we record the frequency infor-

[2] In each document, the first sentence gets the highest score, and the last one the lowest.

[3] http://opennlp.sourceforge.net/

mation (freq_x) for each n-gram from the entity type corpus. This frequency count is used to score the sentences in the input documents (i.e., the web documents which are input to the summarizer for automatic summarization).

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. Song and Croft (1999), for example, used n-gram language models in a generative paradigm and first derived a distinct n-gram language model for each document. Based on this language model, the probability of generating each term in the query is computed. The probability of generating the query is the product of probabilities of generating each of the 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, then this document is more relevant to the query.

Nenkova, Vanderwende, and McKeown (2006) investigated the impact of generative language models on multidocument summarization and compared such models to a nongenerative approach. In their experiments, they used DUC data for development and testing: They used the DUC 2003 input documents for generating their language model and tested the impact of the model on the DUC 2004 data. The language model (M) contains single words with probabilities obtained through corpus statistics, $p(w_j) = C_{wj}/N$, where C_{wj} is the number of times the word w_j occurs in the corpus and where N is the total count of words in the corpus. Nenkova et al. used the language model M to score each sentence S in the summarizer input documents based on two different approaches: accumulative and generative.

$$SumScore(S, M) = \sum_{w_j \in S} \mathrm{p}_M(\mathrm{w}_j). \tag{2}$$

$$AverageScore(S, M) = \frac{\sum_{w_j \in S} \mathrm{p}_M(\mathrm{w}_j)}{|\{\mathrm{w}_j \,|\, \mathrm{w}_j \in S\}|}. \tag{3}$$

$$MultiScore(S, M) = \prod_{w_j \in S} p_M(\mathrm{w}_j). \tag{4}$$

In the accumulative scoring, the authors used the sum of word probabilities obtained from model M to score each sentence of the input documents. This is done both with normalization over the total number of words in a sentence (Equation [3]) and without such normalization (Equation [2]). Instead of using probability values, the actual frequencies of the words could be used to compute these accumulative scores. The accumulative score computation (i.e., summation) is not affected by whether a frequency or a probability or another representation is used. However, this is not the case in a generative scenario, where the likelihood of a sentence being generated by model M is computed, as given in Equation (4). According to Equation (4), short sentences are given higher likelihood than are long ones regardless of their summary relevance because the probability values are always between 0 and 1, so their product will be greater in cases of shorter sentences than in cases of longer ones because of the nature of multiplication with numbers from this interval: the more factors in the multiplication, the smaller the product. Nenkova et al. (2006) evaluated the quality of their summaries using ROUGE (Lin, 2004). Compared to other summarization systems whose performances also are reported on the same DUC 2004 data, the summaries generated by Nenkova et al. (2006) through Equations (2), (3) and (4) are ranked 4, 6, and 16, respectively. In total, there are 20 different systems (including the ones of Nenkova et al.).

We use n-gram language models as a second method for representing entity type models. We use these models in generative way; however, we address the problem of the unfair bias of short sentences over long ones and use the geometric mean of the computed probability score over the entire sentence.

Application to Entity Type Corpora

As an alternative to signature words, we also generated language models from the entity type corpora. As in the case of signature word models (see Figure 3), we generate a unigram and a bigram model from each entity type corpus:

$$\begin{aligned} ngramModel = (&\text{entityType}[(\text{ngram}_1, \text{prob}_1), \ldots, \\ &(\text{ngram}_n, \text{prob}_n)]), \end{aligned} \tag{5}$$

where again ngram_x is either a unigram or a bigram. However, instead of taking the raw frequency counts of each n-gram, as in the signature words model (see Figure 3), we use probabilities for n-gram language models. The probability (prob_x) of an n-gram is calculated using the Good-Turing estimation (Jurafsky & Martin, 2008).

Dependency Patterns

Dependency patterns are concatenated terms extracted from dependency parse trees. Like signature words and language models, dependency patterns have been exploited in various language-processing applications. In information extraction, for instance, dependency patterns have been used to fill manually constructed domain templates with information extracted from text resources (Bunescu & Mooney, 2005; Culotta & Sorensen, 2004; Stevenson & Greenwood, 2005, 2009; Sudo, Sekine, & Grishman, 2001; Yangarber, Grishman, Tapanainen, & Huttunen, 2000), and also to automatically create these domain templates (Banko & Etzioni, 2008; Etzioni, Banko, Soderland, & Weld, 2008; Filatova, Hatzivassiloglou, & McKeown, 2006; Li, Jiang, & Wang, 2010; Sekine, 2006; Sudo, Sekine, & Grishman, 2003).

However, dependency patterns have not been used extensively in summarization tasks. We are only aware of the work described in Nobata, Sekine, Isahara, and Grishman (2002), who used dependency patterns in combination with other features to generate extracts in a single document summarization task. They used the DUC 2001 training set to derive their patterns. The set contains 30 topics, each with 10 documents. For each topic, their patterns are derived by first parsing the sentences in the topic documents for dependency analysis and later extracting the most frequent dependency subtrees from them. In testing, they parse each sentence in the same way that they do for the training sentences, derive patterns from it, and check whether these patterns occur in the set of patterns obtained from the training data. For each match, they take the accumulated frequency information of the training patterns to score the sentence. The authors did not separately report the performance of each feature on the quality of the summaries. However, they mentioned that when learning weights in a simple feature weighting scheme, the weight assigned to dependency patterns was lower than that assigned to other features. The small contribution of the dependency patterns may have been due to the small number of documents that they used to derive their dependency patterns; as mentioned earlier, they gathered dependency patterns from only 10 domain-specific documents which are unlikely to be sufficient to capture repeated features in a domain.

Application to Entity Type Corpora

We use our entity type corpora to derive dependency patterns. Our patterns are derived from dependency trees which are obtained using the Stanford parser.[4] Each article in each entity type corpus was preprocessed by sentence splitting and named entity tagging.[5] Then, each sentence was parsed by the Stanford dependency parser to obtain relational patterns. As with the chain model introduced by Sudo et al. (2001), our relational patterns are concentrated on the verbs in the sentences and contain $n + 1$ words (the verb and n words in direct or indirect relation with the verb). The number n was experimentally set to two words.

For illustration, consider the sentence shown in Table 3 that is taken from an article in the *bridge* corpus. The first two rows of the table show the original sentence and its form after named entity tagging. As in signature words and language models, the next step in processing is to replace any occurrence of a string denoting the entity type by the term "entityType," as shown in the row 3 of Table 3. The final two rows of the table show the output of the Stanford dependency parser and the relational patterns identified for this example.

TABLE 3. Example sentence for dependency pattern.

Original sentence	The bridge was built in 1876 by W. W.
After namedEntity tagging	The bridge was built in DATE by PERSON
Input to the parser	The entityType was built in DATE by PERSON
Output of the parser	det(entityType-2, The-1), nsubjpass(built-4, entityType-2), auxpass(built-4, was-3), prep-in(built-4, DATE-6), agent(built-4, PERSON-8)
Patterns	The entityType built, entityType was built, entityType built DATE, entityType built PERSON, was built DATE, was built PERSON

TABLE 4. Five frequent patterns from the entity type corpora *river* and *volcano*.

River	Volcano
location is entityType, is a tributary, length is km, is entityType flows, location is located	location is entityType, is entityType located, is active entityType, is complex entityType, is highest entityType

To obtain the relational patterns from the parser output, we first identified the verbs in the output. For each such verb, we extracted two further words being in direct or indirect relation to the current verb. Two words are directly related if they occur in the same relational term. The verb "built-4," for instance, is directly related to DATE-6 because they both are in the same relational term "prep-in(built-4, DATE 6)." Two words are indirectly related if they occur in two different terms, but are linked by a word that occurs in those two terms. The verb "was-3" is, for instance, indirectly related to entityType-2 because they are both in different terms, but lare inked with built-4 that occurs in both terms. Note that we consider all direct and indirect relations while generating the patterns. The patterns generated for the example sentence are shown in the bottom of Table 3.

Following these steps, we extracted relational patterns for each entity type corpus along with the frequency of occurrence of the pattern in the entire corpus. Table 4 shows five frequent patterns from the entity type corpora *river* and *volcano*.

Entity Type Model Features

In the previous section, we described three methods for creating entity type models from the entity type corpora. We will use these different models as an *entityTypeModel* feature to compute sentence scores in our summarizer. Depending on which entity type modeling method is used, this feature will be named differently, and its application in computing sentence scores will be different. Next, we describe how sentence scores are computed with each of the entity type model features.

Signature Words

We use the signature words to score each sentence in the input documents according to Equation (6). In the equation,

[4] http://nlp.stanford.edu/software/lex-parser.shtml

[5] For performing shallow text analysis including named entity tagging, the OpenNLP tools were used.

the score of a sentence S is the sum of frequencies (*freq*) of n-grams from the signature word model *SigM* also found in sentence S. We refer to this feature as *SigMSim.*[6]

$$SigMSim(S, SigM) = \sum_{ngram \in SigM \cap S} freq_{SigM}(\text{ngram}). \tag{6}$$

Language Models

The sentence score with language models is calculated according to Equation (7).

$$LMSim(S, LM) = \sqrt[n]{\prod_{ngram \in S} prob_{LM}(ngram)}. \tag{7}$$

In this case, the score of sentence S is the product of probabilities (*prob*) of its n-grams where the prob values are obtained from the language model *LM*. We refer to this feature as LMSim.[7]

We take the geometric mean of the generative model shown in Equation (3) (where n is the number of n-grams constructed from sentence S). This is to avoid the problem of favoring short sentences over long ones by the generative model, as discussed earlier.

Dependency Patterns

The score with the dependency patterns is computed in a similar fashion to the *SigMSim* feature. We assign each sentence a dependency similarity score. To compute this score, we first parse the sentence on the fly with the Stanford parser and obtain the dependency patterns for the sentence. We then associate each dependency pattern of the sentence with the occurrence frequency of that pattern in the dependency pattern model (*DpM*). The dependency pattern feature (*DpMSim*) is then computed as given in Equation (8). It is the sum of all occurrence frequencies of the dependency patterns in the *DpM* detected also in sentence S.

$$DpMSim(S, DpM) \sum_{p \in S} freq_{DpM}(p). \tag{8}$$

Dependency Patterns for Redundancy Reduction and Sentence Ordering

Apart from sentence scoring, the dependency patterns also can be used to address two further challenges of multi-document summarization: the reduction of redundancy and sentence ordering. In this section, we outline and evaluate a possible way that dependency patterns could be used for these tasks.

We can use the dependency pattern approach to address the problem of redundancy in the output summary in a novel way. Often, important information which must be included in the summary is repeated several times across the document set, but must be included in the summary only once. The common approach to avoiding redundancy is to use a text-similarity measure to block the addition of a further sentence to the summary if it is too similar to one that is already included. Instead, since specific dependency patterns express specific types of information, we can group the patterns into groups expressing the same type of information and then, during sentence selection, ensure that sentences matching patterns from different groups are selected to guarantee broad, nonredundant coverage of information relevant for inclusion in the summary. This means that we may want to ensure that the summary contains a sentence describing the type of the entity, its location, and some background information. For example, for the entity *Eiffel Tower*, we may aim to say that it *is a tower*, *located in Paris*, *designed by Gustave Eiffel*, *has a height of 324 m*, and so on. To be able to do so, we categorize dependency patterns according to the type of information that they express:

- **Entitytype:** sentences containing the "entity type" information of the entity (e.g., *Eiffel Tower is a tower*)
- **Location:** sentences containing information about where the entity is located
- **foundationyear:** sentences containing information about when the entity was built
- **specific:** sentences containing some specific information about the entity
- **surrounding:** sentences containing information about what other entities are close to the main entity
- **visiting:** sentences containing information about, for example, visiting times, and so on.

We manually assigned each dependency pattern in each corpus-derived model to one of the attributes just mentioned, provided it occurred five or more times in the entity type corpora. The patterns extracted for our example sentence shown in Table 3, for instance, are all categorized by foundation-year attribute because all of them contain information about the foundation date of an entity.

We make use of these attributes and apply the dependency patterns to categorize the sentences from the input documents to reduce the redundancy and order sentences within the summary. We refer to these summaries as *DepCat* summaries. Note that *DepCat* uses dependency patterns to categorize the sentences rather than rank them. It can be used independently from other features to categorize each sentence by one of the attributes described earlier. To do this, we obtain the relational patterns for the current sentence, check whether each such pattern is included in the *DpM*, and, if so, add the attribute that the pattern was manually associated with to the sentence.

For *DepCat*, we proceed as follows. We first categorize the sentences into the six information types specified earlier. We sort the sentences in each category according to their

[6] We use SigMSim-1 to refer to unigram signature models, and SigMSim-2 to bigram ones.

[7] We use LMSim-1 to refer to unigram language models, and LMSim-2 to bigram ones.

sentence scores. The best scoring sentence goes to the top. Then, we select from the categories (starting from top of the ranked list) sentences in the summary until the summary limit of 200 words is reached. We select the sentences from the categories in the order of: "entityType," "location," "foundationyear," "specific," "surrounding," and "visiting." From each of the first three categories ("entityType," "location," and "foundationyear"), we take a single sentence to avoid redundancy. The same is applied to the final two categories ("surrounding" and "visiting"). Then, if the length limit is not violated, we fill the summary with sentences from the "specific" category until the word limit of 200 words is reached.

Summarizer

Our summarizer is an extractive, query-based, multi-document summarization system. It is given two inputs: a geo-located entity name and a set of documents to be summarized which have been retrieved from the web using the entity name as a query. The summarizer creates image descriptions in a four-step process. First, it applies shallow text analysis, including sentence detection, tokenization, lemmatization, and POS-tagging, to the given input documents. Next, it extracts features from the document sentences and then combines the features using a linear weighting scheme to compute the final score for each sentence. Finally, it composes the final summary using the scored sentences. The following subsections describe these steps in more detail.

Feature Extraction

Within our summarizer we use the following features:

- **qSim:** Sentence similarity to the query, computed as the cosine similarity over the vector representation of the sentence and the query. Each vector position contains tf–idf (Manning et al., 2008; Salton & Buckley, 1988) scores for the words. The idf table is generated on the fly from the input web documents.
- **cenSim:** Sentence similarity to the centroid, computed as cosine similarity over the vector representation of the sentence and the centroid. As in Radev et al. (2004), we keep in each document vector only the 100 words in the document containing the highest tf–idf score.
- **senPos:** Position of the sentence within its document. The first sentence in the document gets the score 1 and the last one gets $\frac{1}{k}$ where k is the number of sentences in the document.
- **isStarter:** A sentence gets a binary score if it starts with the query (geo-located entity name) term (e.g., *Westminster Abbey, The Westminster Abbey, The Westminster,* or *The Abbey*) or with the entity type (e.g., *The church*). We also allow gaps (up to four words) between "the" and the query to capture cases such as *The most magnificent abbey*, and so on.
- **entityTypeModel:** A sentence is scored according to a entity type model derived from Wikipedia entity type corpora.

Sentence Scoring

To compute the final score for each sentence, we use a linear function with weighted features:

$$S_{score} = \sum_{i=1}^{n} (\text{feature}_i \times \text{weight}_i). \qquad (9)$$

To obtain the feature weights for sentence scoring, we use linear regression. Linear regression is a least square error method. It finds the values for the feature weights by predicting the actual sentence scores using the values of the sentence features. Because of this, it requires some training data consisting of assessed sentences, where each sentence has a final score and values for the features.

Our training data contain for each image a set of image descriptions taken from the *VirtualTourist*[8] travel community website. We took all existing image descriptions about a particular image or entity. Note that some of these descriptions about a particular entity were used to derive the model summaries for that entity (for data, see next section). Assuming that model summaries contain the most relevant sentences about an entity, we perform ROUGE comparisons between the sentences in all the image descriptions and the model summaries (i.e., we pair each sentence from all image descriptions about a particular place with every sentence from all the mode summaries for that particular entity). Sentences which are exactly the same or have common parts will score higher in ROUGE than will sentences which do not have anything in common. In this way, we have for each sentence from all existing image descriptions about an entity a ROUGE score indicating its relevance. For each training sentence, we also extract different combination of features. From the set of our features (nine total, including four standard features and five entity type model features), we perform combinations consisting of only two features, three features, four features, and continue up to nine features. For each combination, we train the feature weights using linear regression. Given the weights, Equation (9) is used to compute the final score for each sentence. The final sentence scores are used to sort the sentences in descending order.

Summary Composition

After the sentence-scoring process, the summarizer selects sentences for summary generation. The summary is constructed by first selecting the sentence that has the highest score, followed by the next sentence with the second-highest score, until the compression rate is reached. As in Saggion and Gaizauskas (2004) and Saggion (2005), 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. The summarizer first eliminates closed-class words (prepositions, articles)

[8] http://www.Virtualtourist.com

TABLE 5. ROUGE scores for each single feature and Wikipedia baseline. The numbers 1 and 2 after the model features *SigMSim* and *LMSim* indicate the use of a unigram (1) or a bigram (2) version of those models.

ROUGE	cenSim	senPoS	qSim	isStarter	SigMSim-1	SigMSim-2	LMSim-1	LMSim-2	DpMSim	Wiki
R2	.0734	.066	.0774	.0869	.08	.079	.079	.0895	.093	.097
RSU4	.12	.11	.12	.137	.133	.133	.135	.142	.145	.14

from the sentences and then measures lemma overlap with the lemmas of the remaining open-class words (nouns, verbs, adjectives, and adverbs), which are, according to Ye, Qiu, Chua, and Kan (2005), a strong basis for measuring similarities between sentences. We refer to this method as *greedySelection*. Note that we do not use *greedySelection* when the *DepCat* feature is used.

Evaluation

To evaluate our approach, we used two different assessment methods: ROUGE (Lin, 2004) and a manual readability assessment.

Data Sets

For evaluation, we use the image collection described in Aker and Gaizauskas (2010). The image collection contains 310 different images with manually assigned entity names. The images cover 60 of the 107 entity types identified from Wikipedia (see Table 2). For each image, there are up to four short descriptions or model summaries. The model summaries were created manually based on image descriptions taken from *VirtualTourist* and contain a minimum of 190 and a maximum of 210 words. Two thirds of this image collection was used to train the weights, and the temaining one third (105 images) was used for evaluation.

To generate automatic captions for the images, we automatically retrieved the top-10 related web documents for each image using the Yahoo! search engine and the entity name associated with the image as a query. The text from these documents was extracted using an HTML parser and passed to the summarizer. The set of documents we used to generate our summaries excluded any *VirtualTourist* related sites, as these were used to generate the model summaries.

ROUGE Assessment

In the first assessment, we compared the automatically generated summaries against model summaries written by humans using ROUGE (Lin, 2004). Following DUC evaluation standards, we used ROUGE 2 (R2) and ROUGE SU4 (RSU4) as evaluation metrics. R2 computes the number of bigram overlaps between the automatic and model summaries. RSU4 allows bigrams to be composed of noncontiguous words, with a maximum of four words between the bigrams.

As baselines for evaluation, we used summaries extracted from the top document retrieved from the web and Wikipedia.

To create the baseline using the top document retrieved from the web, we use the geo-located entity names to automatically query related documents from the web using the Yahoo! Search engine. For each entity name, we take the top-ranked non-Wikipedia document retrieved in the Yahoo! search results and generate a baseline summary by selecting sentences from the beginning until the summary reaches a length of 200 words.

The Wikipedia baseline summaries are generated using the Wikipedia article for a given geo-located entity. From this article, we again select sentences from the beginning until the summary length of a limit of 200 words is reached. For each geo-located entity, the corresponding Wikipedia article was manually identified from the list of documents retrieved by the Yahoo! Search engine. By doing this, we ensured that we took the correct Wikipedia article.

By using both the first document and Wikipedia baselines, we simulate the scenario in which image descriptions are generated by a simple web search, without needing the summarizer. In other words, our system needs to significantly outperform these baselines to justify using multi-document summarization for image captioning.

We consider the top-ranked non-Wikipedia document to be a weaker baseline than is a Wikipedia article, which we take to be a strong baseline against which to compare the automated summaries. Wikipedia articles focus only on the topic that they were written about whereas an arbitrary non-Wikipedia web document may contain other unrelated information.

First, we compared the baseline summaries against the *VirtualTourist* model summaries. Wikipedia baseline ROUGE scores (R2 .097***, RSU4 .14***) are significantly higher than the first or top-document ones (R2 0.042, RSU4 .079).[9]

Second, we separately ran the summarizer over the input web documents for each single feature and compared the automated summaries against the model ones. The results of this comparison are shown in Table 5.

From the table, we see that automated summaries using each of the features achieved lower ROUGE scores than did

[9] To assess the statistical significance of ROUGE score differences between multiple summarization results, we performed a pairwise Wilcoxon signed-rank test. We use the following conventions for indicating significance level: ***$p < .0001$. **$p < .001$. *$p < .05$. No star = nonsignificance.

DOI: 10.1002/asi

TABLE 6. Model, Wikipedia baseline, and isStarter + LMSim-2 + DepCat summary for Eiffel Tower.

Model summary	Wikipedia baseline summary	isStarter + LMSim-2 + DepCat summary
The Eiffel Tower is the most famous place in Paris. It is made of 15,000 pieces fitted together by 2,500,000 rivets. It's of 324 m (1,070 ft) high structure and weighs about 7,000 tons. This world famous landmark was built in 1889 and was named after its designer, engineer Gustave Alexandre Eiffel. It is now one of the world's biggest tourist places which is visited by around 6,5 million people yearly. There are three levels to visit: Stages 1 and 2 which can be reached by either taking the steps (680 stairs) or the lift, which also has a restaurant "Altitude 95" and a Souvenir shop on the first floor. The second floor also has a restaurant "Jules Verne". Stage 3, which is at the top of the tower can only be reached by using the lift. But there were times in the history when Tour Eiffel was not at all popular, when the Parisians thought it looked ugly and wanted to pull it down. The Eiffel Tower can be reached by using the Mtro through Trocadro, Ecole Militaire, or BirHakeim stops. The address is: Champ de MarsTour Eiffel.	The Eiffel Tower (French: Tour Eiffel, [tur efel]) is a 19th century iron lattice tower located on the Champ de Mars in Paris that has become both a global icon of France and one of the most recognizable structures in the world. The Eiffel Tower, which is the tallest building in Paris, is the single most visited paid monument in the world; millions of people ascend it every year. Named after its designer, engineer Gustave Eiffel, the tower was built as the entrance arch for the 1889 World's Fair. The tower stands at 324 m (1,063 ft) tall, about the same height as an 81 story building. It was the tallest structure in the world from its completion until 1930, when it was eclipsed by the Chrysler Building in New York City. Not including broadcast antennas, it is the second-tallest structure in France, behind the Millau Viaduct, completed in 2004. The tower has three levels for visitors. Tickets can be purchased to ascend either on stairs or lifts to the first and second levels.	The Eiffel Tower, which is the tallest building in Paris, is the single most visited paid monument in the world; millions of people ascend it every year. (**entityType**) The tower is located on the Left Bank of the Seine River, at the northwestern extreme of the Parc du Champ de Mars, a park in front of the Ecole Militaire that used to be a military parade ground. (**location**) The tower was met with much criticism from the public when it was built, with many calling it an eyesore. (**foundationyear**) Counting from the ground, there are 347 steps to the first level, 674 steps to the second level, and 1,710 steps to the small platform on the top of the tower. Although it was the world's tallest structure when completed in 1889, the Eiffel Tower has since lost its standing both as the tallest lattice tower and as the tallest structure in France. The tower has two restaurants: Altitude 95, on the first floor 311ft (95 m) above sea level; and the Jules Verne, an expensive gastronomical restaurant on the second floor, with a private lift. (**specific**) There is an entrance fee of between euro;4.10 and euro;10.70 for adults and between euro;2.30 and euro;5.90 for children, depending on which floor you wish to visit by elevator. (**visiting**)

TABLE 7. ROUGE scores of feature combinations which score moderately or significantly higher than does the dependency pattern model (DpMSim) feature and the Wikipedia baseline.

ROUGE	isStarter + LMSim-2	isStarter + LMSim-2 + DepCat***	DpMSim	Wiki	User-to-user
R2	.095	.102	.093	.097	0.11
RSU4	.145	.155	.145	.14	0.16

the Wikipedia baseline, thus indicating that initial sentences from Wikipedia articles are indeed of high quality. The opposite is true for the summaries obtained from the first top web document: The automated summaries using any of our summarization features scored higher than did the first document baseline ones (R2 .042, RSU4 .079, not shown in the table). For this reason, we will focus on the Wikipedia baseline summaries to draw conclusions about the quality of our automatic summaries. Table 6 shows the Wikipedia baseline summary for the *Eiffel Tower*.

Turning to the ROUGE results for single summarization features in Table 5, we can see that the dependency model feature (*DpMSim*) contributes most to the summary quality according to the two ROUGE metrics. It achieved significantly higher ROUGE scores than did all other features (***), except the *LMSim-2* feature, where it led to a small improvement. Compared to the Wikipedia baseline (*Wiki*), the *DpMSim* summaries achieved insignificantly different ROUGE scores.

The lowest ROUGE scores are obtained if only sentence position (*senPos*) is used. These scores are significantly lower than those of the Wikipedia baseline, which also is true for all other features except *LMSim-2* and *DpMSim*.

To see how the ROUGE scores change when features are combined with each other, we used different combinations of the features, ran the summarizer for each combination, and compared the automated summaries against the model ones[10] Among the different combinations, we also included the dependency pattern categorization (*DepCat*) feature.[11] Table 7 shows the results of feature combinations which score moderately or significantly higher than the dependency pattern model (*DpMSim*) feature score shown in

[10] For each feature combination, a different set of weights are trained using linear regression.

[11] *DepCat* is used to reorder the sentences scored by other features. It is not included in Equation (9) to obtain a feature combination. In addition, when *DepCat* is used, we switch off the *greedySelection*.

TABLE 8. Readability evaluation results: Wikipedia baseline (W), isStarter + LMSim–2 (SLM) and isStarter + LMSim-2 + DepCat (SLMD).

	5			4			3			2			1		
Criterion	W	SLM	SLMD	W	SLM	SLMD	W	SLM	SLMD	W	SLM	SLMD	W	SLM	SLMD
Clarity	72.6	50.5	53.6	21.7	30.0	31.4	1.2	6.7	5.7	4.0	10.2	6.0	0.5	2.6	3.3
Coherence	67.1	39.0	48.3	23.6	31.4	26.9	4.8	12.4	11.9	3.3	10.2	9.8	1.2	6.9	3.1
Focus	72.1	49.3	51.2	20.5	26.0	25.2	3.8	10.0	10.7	3.3	10.0	10.5	0.2	4.8	2.4
Grammar	48.6	55.7	62.9	32.9	29.0	30.0	5.0	3.1	1.9	11.7	12.1	5.2	1.9	0	0
Redundancy	69.8	42.9	55.0	21.7	17.4	28.8	2.4	4.5	4.3	5.0	27.1	8.8	1.2	8.1	3.1

Table 5. In Table 7, we also give ROUGE scores of model summaries compared to each other (column *User-To-User*), which represent the upper bound scores one could achieve with automatic summaries.

The results show that combining *DpMSim* with other features did not lead to higher ROUGE scores than those produced by that feature alone. In contrast, the feature *LMSim-2*, which on its own has a performance insignificantly different from *DpMSim* (Table 5), combines well with other features. In combination with the *isStarter* feature, it achieved ROUGE scores comparable to *DpMSim*. The best results, however, are achieved if categorization using dependency patters (*DepCat*) is added to this combination (*isStarter + LMSim-2 + DepCat*). Such summaries categorized by dependency patterns achieved significantly higher ROUGE scores than did the Wikipedia baseline[12] and also were very close to the User-to-User upper bound scores. Table 6 shows a summary about the *Eiffel Tower* obtained using this *isStarter + LMSim-2 + DepCat feature*.

Readability Assessment

We also evaluated our summaries using a readability assessment as in the Document Understanding Conference and the Text Analysis Conference (TAC). The DUC and the TAC manually assess the quality of automatically generated summaries by asking human subjects to score each summary using five criteria: *grammaticality, redundancy, clarity, focus,* and *structure*. Each criterion is scored on scale of 1 (*strongly disagree*) to 5 (*strongly agree*), with high scores indicating a better result (Dang, 2005).

For this evaluation, we used the same 105 entities as in the ROUGE evaluation. As the ROUGE evaluation showed that the dependency pattern categorization (*DepCat*) renders the best results when used in the feature combination *isStarter + LMSim-2 + DepCat*, we also performed the readability assessment on summaries generated using this feature combination. For comparison, we also evaluated summaries which were not structured by dependency patterns (*isStarter + LMSim-2*) and the Wikipedia baseline summaries.

We asked four people to assess the summaries. Each person was shown all 315 summaries (105 from each summary type) in a random way and was asked to assess them according to the DUC and the TAC manual assessment scheme (Dang, 2005, 2006). The results are shown in Table 8. In the table, each cell shows the percentage of summaries scoring the ranking score heading the column for each criterion in the row, as produced by the summary method indicated by the subcolumn heading. The numbers indicate the percentage values averaged over the four assessors.

From Table 8, we see that using dependency patterns to categorize the sentences and produce a structured summary helps to obtain more readable summaries. Looking at the 5 and 4 scores, the table shows that the dependency-pattern categorized summaries (*SLMD*) have better clarity (85% of the summaries), are more coherent (74% of the summaries), contain less redundant information (83% of the summaries), and have better grammar (92% of the summaries) than do the ones without dependency categorization (80, 70, 60, 84%, respectively). The large difference in redundancy scores (83 vs. 60%) shows in particular that the *DepCat* feature is a useful feature for redundancy reduction in summaries.

The scores of our automated summaries were better than those of the Wikipedia baseline summaries in the grammar feature. We included the grammar feature in the evaluation to be consistent with the evaluation criteria used in the DUC and the TAC. For extractive summarization, however, the grammar feature is not relevant, as it can be assumed that extracted sentences are fully grammatical. In all other relevant features, the Wikipedia baseline summaries obtained better scores than did our automated summaries. This comparison shows that there is still a gap to fill to obtain more readable summaries.

Discussion

In our single-feature analysis, the results indicate that the entity type model features indeed help the summarizer to produce better summaries. Using any of our entity type model features, we have obtained higher ROUGE scores than when standard summarization features *cenSim*, *senPos*, and *qSim* were used to produce the summaries. However, not all methods for entity type modeling have shown equal performance, suggesting that the way entity type models are represented plays a role in how useful they are as summarization features. In our case, summaries obtained through the standard feature *isStarter* are better than those generated by signature word (*SigMSim*) and unigram language models

[12] For both ROUGE R2 and ROUGE SU4, the significance level is $p < .0001$.

(*LMSim-1*). The *isStarter* feature looks in each sentence only for an occurrence of the given query (entity name) and entity type. We believe that sentences starting with the query or entity type are likely to be salient for the given entity name, which therefore leads to better scoring summaries. Bigram language models (*LMSim-2*) and dependency patterns (*DpMSim*), on the other hand, significantly outperformed the *isStarter* feature, *DpMSim* being the single feature which led to the highest scoring summaries, almost identical to those of the Wikipedia baseline.

Thus, we can conclude that the summaries obtained using signature word and language models are not as good as the ones obtained using dependency patterns. The main weakness of signature words and n-gram language models is that they only capture very local information about short-term sequences and cannot model long-distance dependencies between terms. For example, one common and important feature of entity descriptions is the simple specification of the entity type (e.g., the information that the entity *London Bridge is a bridge* or *that the Rhine is a river*). If this information is expressed as in the first line of Table 9, signature words and n-gram language models are likely to reflect it since one would expect the trigram *is a bridge* to occur with high frequency in a corpus of bridge descriptions. However, if the type predication occurs with less commonly seen local context, as is the case for the entity *Rhine* in the second row of Table 9—*one of the longest and most important rivers*—signature words and n-gram language models may well be unable to identify it.

Intuitively, what is important in both these cases is that there is a predication whose subject is the entity instance of interest, and the head of whose complement is the entity type: *London Bridge . . . is . . . bridge* and *Rhine . . . is . . . river*. Sentences matching such patterns are likely to be important ones to include in a summary. The results suggest that rather than representing entity type models via corpus-derived signature words or language models, it is better to represent them using corpus-derived dependency patterns.

The investigation of feature combinations also has shown that using dependency patterns for redundancy reduction and sentence ordering within a summary (feature *DepCat*) significantly improves the quality of summaries. Interestingly, when dependency patterns are used for sentence scoring (*DpMSim*), no further improvement could be observed in additionally using dependency patterns for redundancy reduction and sentence ordering (*DepCat*). However, *DepCat* significantly improved the ROUGE scores of the summaries generated by the combination of the bigram language models (*LMSim-2*) and the *isStarter* feature (*isStarter* + *LMSim-2* + *DepCat*). This combination of features produced structured summaries which led to significantly better results than those with Wikipedia baseline summaries and were almost equal to human-generated baseline summaries as assessed by ROUGE. Human readability assessment reflected these ROUGE scores for the grammaticality aspect of the summaries. However, the automated *isStarter* + *LMSim-2* + *DepCat* summaries scored lower in fluency and redundancy than did the Wikipedia baseline, indicating that usage of *DepCat* for these purposes still has scope for improvement.

TABLE 9. Example of sentences which express the type of an entity.

London Bridge is a bridge
The Rhine (German: Rhein; Dutch: Rijn; French: Rhin; Romansh: Rain; Italian: Reno; Latin: Rhenus West Frisian Ryn) is one of the longest and most important rivers in Europe

From these results, we can conclude that it is possible to generate higher quality geo-located entity descriptions using automatic summarization techniques than by simply referring to the existing descriptions in Wikipedia, which justifies using automatic summarization for image description generation generally and not only in cases where no Wikipedia descriptions for a given entity exist. Since use of entity type models represented as dependency patterns was crucial for achieving this result, we conclude that dependency patterns are worth investigating for entity-focused automated-summarization tasks. Such investigations should in particular concentrate on how dependency patterns can be used to order sentences within the summary, as our best results were achieved when dependency patterns were used for this purpose. In particular, replacing manual categorization of dependency patterns, which was necessary for this purpose, with an automatic procedure needs to be addressed.

Finally, note that our testing set contains very popular geo-located entities which are famous tourist attractions and are described in Wikipedia. In practice, one could use the first paragraph of the associated Wikipedia article about such an entity as a summary and not automatically generate one. However, this is not possible if a geo-located entity does not have a Wikipedia article. Our results show that in this case, an automated summary is indeed a good option. However, one also could argue that the number and the quality of web documents related to a less popular entity—that is, an entity for which Wikipedia does not have an entry—will decrease, and these factors might affect the quality of the automated summaries. Although this remains to be experimentally tested, in Aker, Fan, Sanderson, and Gaizauskas (2012), we showed that automated summaries even for less popular geo-located entities are useful for image indexing and retrieval. In that work, we used about 6,000 images downloaded from Flickr.com and evaluated summaries generated by different summarization techniques in the image-retrieval-effectiveness task. We also used existing Flickr textual information as a baseline. We showed that combining the Flickr texts with entity type model biased summaries performs significantly better, as compared to all other index types (i.e., existing Flickr captions and summaries generated without entity type models).

Grouping of Geo-Located Entity Types

Aker, Plaza, and Lloret (2013) and Aker and Gaizauskas (2011) investigated which information types (attributes) humans associate with geo-located entities from urban and rural landscapes. Both these studies identified a set of attributes that are relevant for any entity type, but also found that an appreciable proportion of attributes is entity type specific. These two studies also showed that if entity types have similar looks and purposes, people tend to agree on what attributes to associate with them. The question now arises whether it is possible to derive entity type models for grouped types, rather than for single types, such that these models still improve the performance of summary generation for a single geo-located entity. This would be very useful when there is a geo-located entity for which not enough or no textual resources are available. In this case, text resources of similar entity types could be used to derive an entity type model for that type. For example, the authors of both studies showed that entity types *church, basilica, abbey, cathedral*, and *temple* correlate highly with each other, meaning that they do share many attributes. Some of these entity types, like *church*, have more frequently occurring instances than do others (e.g., *basilica*); that is, there are typically more churches than basilicas, and therefore it can be expected that there are more church descriptions from which to build entity type corpora than there are basilica descriptions. If a summary for a basilica needs to be generated but little or no information exists on this entity, then texts describing churches and other religious geo-located buildings could be used to derive entity type models, and these models can be used to generate a description of the basilica in question. We therefore investigated whether deriving entity type models from grouped entity type corpora has any affect on the summary results.

In total, we have 60 entity types (discussed earlier). One could apply machine-learning techniques to perform hierarchical grouping between them. However, for simplicity, we perform manual grouping based on the look and purpose of the entity types. The resulting set of groups of similar entity types is shown in Table 10.

TABLE 10. Groups of entity types.

Group name	Entity types within the group
Religious places	Church, cathedral, chapel, basilica, synagogue, abbey, shrine, mosque, temple
Mountainous areas	Mountain, peak, volcano, ski resort, glacier, hill
Buildings	Tower, skyscraper, house, building, residence, palace, castle, hotel, parliament
Water bodies	Canal, lake, river, waterfall
Cultural attractions	Museum, opera house, gallery
Roads	Road, avenue, boulevard
Streets	Street, square
Transport sites	Railway, railway station
Seasides	Beach, coast, bay
Populated áreas	District, village, city
Education	College, university
Shopping areas	Market, shopping center, shop, store
Monuments	Monument, statue
Places of entertainment	Restaurant, casino, bar, cinema, pub, club
Civil engineering	Bridge, gate, arch
Places for relaxation	Park, garden
Places for sport	Stadium, arena
Animal theme parks	Zoo, aquarium

Using these groups of entity types, we derive entity type models. We investigate only the bigram language model (*LMSim-2*) and the dependency model (*DpMSim*) because they were the best performing features in the previous experiment. With this, we aimed to investigate whether deriving these two models from grouped entity type corpora has any affect on the summary results. The results of the ROUGE evaluation are shown in Table 11.

TABLE 11. ROUGE scores of features LMSim-2, LMSim-2g, DpMSim, and DpMSim-g[a]

ROUGE	LMSim-2	LMSim-2g	DpMSim	DpMSim-g
R2	.089	.087	.093	.092
RSU4	.142	.14	.145	.144

Note. [a]"g" indicates features which are derived from groups of entity type corpora.

From Table 11, we can see that compared to single entity type models, there is a small decrease in both ROUGE 2 and ROUGE SU4 scores when group of entity types are used to derive the models. These nonsignificant changes on the scores show, in general, that grouping of similar entity types can be performed without losing too much in summary quality. Therefore, if there is an entity type for which not enough or no textual resources are available, text resources of similar entity types could be used to build an entity type model for that type. However, when text resources exist for every single entity type, as is the case in our entity type corpus, the results indicate that deriving single entity type models instead of group models and using these in generating image descriptions lead to better ROUGE results.

Conclusion

In this article, we investigated three different methods to derive entity type models from entity type corpora: signature words, language models, and dependency patterns. We discussed the use of these methods within the summarizer to bias sentence selection. We showed that dependency pattern models yield summaries which score higher than do summaries obtained using signature word or language models which use a simpler representation of an entity type model. Dependency pattern models can contribute both to better sentence scoring and readabilty in particular clarity and coherence scores. Thus, we conclude that entity type models as represented by dependency patterns do lead to improved results in entity-focused, automatic-text summarization. Finally, we also showed that deriving entity type models from groups of similar entity types is possible, which is useful in cases in which there exist limited text resources for single entity types. For such entity types, entity type models of similar entity types can be used instead without losing too much in summary quality.

Future Work

State-of-the-art summaries are obtained when the summary generation is formulated as a search problem (Alfonseca & Rodriguez, 2003; Gillick & Favre, 2009; Gillick, Riedhammer, Favre, & Hakkani-Tur, 2009; Li, Qian, & Liu, 2013; Lin & Bilmes, 2010; Liu, He, Ji, & Yang, 2006; McDonald, 2007; Orasan, 2003; Riedhammer, Gillick, Favre, & Hakkani-Tur, 2008; Woodsend & Lapata, 2012; Yih, Goodman, Vanderwende, & Suzuki, 2007). We plan to adopt one of these search approaches to use in our summarization task. During the search, the inclusion of a sentence into the summary is determined by several constraints such as its length, its similarity to the summary generated so far, and so on. To these constraints, we plan to integrate preexisting sentence-ordering models (Barzilay & Lapata, 2008; Bollegala, Okazaki, & Ishizuka, 2012; Guinaudeau & Strube, 2013; Lapata, 2003; Lin, Ng, & Kan, 2011; Louis & Nenkova, 2012; Soricut & Marcu, 2006). In such works, the task of the sentence-ordering procedure is to have the artificially permuted sentences of a single coherent document returned to their initial order or to compute scores for coherence assessment of already-generated automatic summaries. Since the dependency patterns when used for sentence ordering led to the best performing summaries, we plan to integrate them into these sentence-ordering models and use the models as an additional constraint for inclusion of a sentence into the summary.

References

Aker, A., & Gaizauskas, R. (2009, September). Summary generation for toponym-referenced images using object type language models. Paper presented at the International Conference on Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria.

Aker, A., & Gaizauskas, R. (2010). Model summaries for location-related images. Proceedings of the International Conference on Language Resources and Evaluation (LREC). In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, D. Tapias (Eds), Paris: ELRA.

Aker, A., & Gaizauskas, R. (2011). Understanding the types of information humans associate with geographic objects. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM), In B. Berendt, A. de Vries, W. Fan, C. Macdonald, I. Ounis, I. Ruthven (Eds.) (pp. 1929–1932). New York, USA: ACM.

Aker, A., Cohn, T., & Gaizauskas, R. (2012). Redundancy reduction for multi-document summaries using A* search and discriminative training. 1st Workshop on Automatic Text Summarization of the Future, Castellon, Spain.

Aker, A., Fan, X., Sanderson, M., & Gaizauskas, R. (2012). Investigating summarization techniques for geo-tagged image indexing. Proceedings of the International Conference on European Conference on Information Retrieval (ECIR), In D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C.P. Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum (Eds.) Berlin Heidelberg: Springer.

Aker, A., Plaza, L., & Lloret, E. (2013). Do humans have conceptual models about geographic objects? A user study. Journal of the American Society for Information Science and Technology (JASIST), 64(4), 689–700.

Alfonseca, E., & Rodriguez, P. (2003). Generating extracts with genetic algorithms. Advances in Information Retrieval (vol. 2633, pp. 511–519). Berlin Heidelberg: Springer.

Armitage, L.H., & Enser, P.G. (1997). Analysis of user need in image archives. Journal of Information Science, 23(4), 287–299.

Banko, M., & Etzioni, O. (2008). The tradeoffs between open and traditional relation extraction. Proceedings of ACL-08: Human Language Technologies (HLT) (pp. 28–36). Shroudsburg, PA: Association for Computational Linguistics, .

Barnard, K., & Forsyth, D. (2001). Learning the semantics of words and pictures, in Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, British Columbia, Canada, July 7–14, 2001. IEEE Computer Society 2001 (pp. 408–415).

Barnard, K., Duygulu, P., Forsyth, D., De Freitas, N., Blei, D.M., & Jordan, M.I. (2003). Matching words and pictures. Journal of Machine Learning Research, 3(1), 1107–1135.

Barzilay, R., & Lapata, M. (2008). Modeling local coherence: An entity-based approach. Computational Linguistics, 34(1), 1–34.

Barzilay, R., McKeown, K., & Elhadad, M. (1999). Information fusion in the context of multi-document summarization. Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (pp. 550–557). Shroudsburg, PA: Association for Computational Linguistics.

Baxendale, P. (1958). Machine-made index for technical literature: An experiment. IBM Journal of Research and Development, 2(4), 354–361. IBM.

Bollegala, D., Okazaki, N., & Ishizuka, M. (2010). A bottom-up approach to sentence ordering for multi-document multidocument summarization. Information Processing & Management, 46(4), 89–109.

Bollegala, D., Okazaki, N., & Ishizuka, M. (2012). A preference learning approach to sentence ordering for multi-document summarization. Information Sciences, 217, 78–95.

Bunescu, R., & Mooney, R. (2005). A shortest path dependency kernel for relation extraction. Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (pp. 724–731). Stroudsburg, PA, USA.: Association for Computational Linguistics.

Culotta, A., & Sorensen, J. (2004). Dependency tree kernels for relation extraction. Proceedings of the 42nd meeting of the Association for Computational Linguistics (ACL'04) (pp. 423–429). Barcelona, Spain.

Dang, H.T. (2005). Overview of DUC 2005. Proceedings of the Document Understanding Conference (DUC).

Dang, H.T. (2006). Overview of DUC 2006. Proceedings of the Document Understanding Conference (DUC).

Deschacht, K., & Moens, M.-F. (2007). Text analysis for automatic image annotation. Proceedings of the 45th annual meeting of the Association of Computational Linguistics (pp. 1000–1007). Shroudsburg, PA: Association for Computational Linguistics.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. Computational linguistics (Vol. 19(1), pp. 61–74). Cambridge, MA, USA: MIT Press.

Duygulu, P., Barnard, K., de Freitas, J., & Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In A. Heyden, G. Sparr, M. Nielsen & P. Johansen (Eds.), Seventh European Conference on Computer Vision (ECCV) (Vol. 4, pp. 97–112), Springer Berlin Heidelberg.

Eakins, J. (1998). Techniques for image retrieval. Library & information briefings, No. 85, British Library Research and Development Department (pp. 1–15).

Etzioni, O., Banko, M., Soderland, S., & Weld, D.S. (2008). Open information extraction from the web. Communications of the ACM (Vol. 51(12), pp. 68–74). New York, NY, USA: ACM.

Eysenck, M., & Keane, M. (2005). Cognitive psychology: A student's handbook. New York NY, USA: Psychology Press.

DOI: 10.1002/asi

Farhadi, A., Endres, I., Hoiem, D., & Forsyth, D. (2009). Describing objects by their attributes. In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR 2009) (pp. 1778–1785).

Farhadi, A., Hejrati, M., Sadeghi, M., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D. (2010). Every picture tells a story: Generating sentences from images. Computer Vision–ECCV 2010 (pp. 15–29). Berlin Heidelberg: Springer.

Feng, Y., & Lapata, M. (2008). Automatic image annotation using auxiliary text information. Proceedings of the meeting of the Association for Computational Linguistics (ACL'2008) (pp. 272–280). Shroudsburg, PA: Association for Computational Linguistics.

Feng, Y., & Lapata, M. (2010a). How many words is a picture worth? automatic caption generation for news images. Proceedings of the 48th annual meeting of the Association for Computational Linguistics (pp. 1239–1249). Shroudsburg, PA: Association for Computational Linguistics.

Feng, Y., & Lapata, M. (2010b). Topic models for image annotation and text illustration. Human Language Technologies: The 2010 annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 831–839). Los Angeles, CA: Association for Computational Linguistics.

Feng, Y., & Lapata, M. (2010c). Visual information in semantic representation. Human Language Technologies: The 2010 annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 91–99). Los Angeles, CA: Association for Computational Linguistics.

Filatova, E., Hatzivassiloglou, V., & McKeown, K. (2006). Automatic creation of domain templates. Proceedings of the International Conference on Computational Linguistics (COLING) on Main conference poster sessions (pp. 207–214). Shroudburg, PA: Association for Computational Linguistics.

Gillick, D., & Favre, B. (2009). A scalable global model for summarization. Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing (pp. 10–18). Association for Computational Linguistics.

Gillick, D., Riedhammer, K., Favre, B., & Hakkani-Tur, D. (2009). A global optimization framework for meeting summarization. In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pp. 4769–4772. IEEE.

Guinaudeau, C., & Strube, M. (2013). Graph-based local coherence modeling. Proceedings of the 51st annual meeting of the Association for Computational Linguistics, pp. 93–103.

Gupta, A., & Mannem, P. (2012). From image annotation to image description. T. Huang, Z. Zeng, C. Li & C. Leung (Eds.), Neural information processing, Vol. 7667 (pp. 196–204). Springer Berlin Heidelberg.

Hovy, E., & Lin, C.-Y. (1998). Automated text summarization and the summarist system. Proceedings of the TIPSTER workshop (pp. 197–214), Baltimore, MD. Association for Computational Linguistics, Stroudsburg, PA, USA.

Jaimes, A., & Chang, S. (2000). A conceptual framework for indexing visual information at multiple levels. IS&T/SPIE Internet Imaging (Vol. 3964, pp. 2–15). Bellingham, WA: SPIE Digital Library.

Jones, K. (1999). Automatic summarizing: Factors and directions. Advances in Automatic Text Summarization (pp. 1–12). Cambridge, MA: MIT Press.

Jorgensen, C. (1998). Attributes of images in describing tasks. Information Processing & Management, 34(2), 161–174. Elsevier.

Jurafsky, D., & Martin, J. (2008). Speech and language processing. India: Prentice Hall.

Kim, I., Le, D., & Thoma, G. (2007). Identification of "comment-on sentences" in online biomedical documents using support vector machines. Proceedings of the SPIE Conference on Document Recognition and Retrieval (Vol. 68150, pp. X1–X9).

Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, C.A., & Berg, T.L. (2011). Baby talk: Understanding and generating simple image descriptions. In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR) (pp. 1601–1608). IEEE.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. Proceedings of the 18th annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 68–73).

Kuznetsova, P., Ordonez, V., Berg, C.A., Berg, T.L., & Choi, Y. (2012). Collective generation of natural image descriptions. Proceedings of the 50th annual meeting of the Association for Computational Linguistics: Long Papers-Vol. 1, pp. 359–368). Association for Computational Linguistics.

Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. Proceedings of the 41st annual meeting on Association for Computational Linguistics (Vol. 1, pp. 545–552). Association for Computational Linguistics.

Li, C., Qian, X., & Liu, Y. (2013). Using supervised bigram-based ilp for extractive summarization. Proceedings of Association for Computational Linguistics (ACL) (pp. 1004–1013).

Li, P., Jiang, J., & Wang, Y. (2010). Generating templates of entity summaries with an entityaspect model and pattern mining. Proceedings of the 48th annual meeting of the Association for Computational Linguistics (pp. 640–649), Association for Computational Linguistics.

Liakata, M., Teufel, S., Siddharthan, A., & Batchelor, C. (2010). Corpora for the conceptualisation and zoning of scientific papers. Proceedings of the International Conference on Language Resources and Evaluation (LREC).

Liddy, E.D. (1991). The discourse-level structure of empirical abstracts: An exploratory study. Information Processing & Management, 27(1), 55–81.

Lin, C. (2004). Rouge: A package for automatic evaluation of summaries. Text summarization branches out: Proceedings of the ACL-04 Workshop (pp. 74–81). Association for Computational Linguistics, Barcelona, Spain.

Lin, C., & Hovy, E. (2000). The automated acquisition of topic signatures for text summarization. Proceedings of the 18th Conference on Computational linguistics (Vol. 1, pp. 495–501). Association for Computational Linguistics.

Lin, C., & Hovy, E. (2002). From single to multi-document summarization: A prototype system and its evaluation. Proceedings of the 40th annual meeting on Association for Computational Linguistics (pp. 457–464). Association for Computational Linguistics.

Lin, H., & Bilmes, J. (2010). Multi-document summarization via budgeted maximization of submodular functions. Human Language Technologies: The 2010 annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 912–920). Los Angeles, CA: Association for Computational Linguistics.

Lin, Z., Ng, H.T., & Kan, M.Y. (2011). Automatically evaluating text coherence using discourse relations. Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies (Vol. 1, pp. 997–1006). Association for Computational Linguistics.

Liu, D., He, Y., Ji, D., & Yang, H. (2006). Genetic algorithm based multi-document summarization. PRICAI 2006: Trends in artificial intelligence (pp. 1140–1144). Berlin Heidelberg: Springer.

Lloret, E., & Palomar, M. (2012). Text summarisation in progress: A literature review. Artificial Intelligence Review, 37(1), 1–41.

Louis, A., & Nenkova, A. (2012). A coherence model based on syntactic patterns (Tech. Report). Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLPCoNLL 2012), Jeju, Korea.

Luhn, H.P. (1958). The automatic creation of literature abstracts. IBM Journal of Research and Development, 2(2), 159–165.

Mani, I. (2001). Automatic summarization (Vol. 3). Amsterdam: John Benjamins Publishing.

Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T., & Sundheim, B. (1999). The tipster summac text summarization evaluation. Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics (pp. 77–85). Association for Computational Linguistics.

Manning, C., Raghavan, P., & Schutze, H. (2008). Introduction to information retrieval, Vol. 1, Cambridge University Press Cambridge.

Marsh, E., & White, M. (2003). A taxonomy of relationships between images and text. Journal of Documentation, 59(6), 647–672.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. Advances in Information Retrieval (pp. 557–564). Berlin Heidelberg: Springer.

Mori, Y., Takahashi, H., & Oka, R. (2000). Automatic word assignment to images based on image division and vector quantization. Proceedings of RIAO 2000: Content-Based Multimedia Information Access.

Nenkova, A., & McKeown, K. (2011). Automatic Summarization. Foundations and Trends in Information Retrieval (Vol. 5, pp. 2–3). Boston: now publishers Inc.

Nenkova, A., Vanderwende, L., & McKeown, K. (2006). A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. Proceedings of the 29th annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 573–580). New York, NY: ACM.

Nobata, C., Sekine, S., Isahara, H., & Grishman, R. (2002). Summarization system integrated with named entity tagging and ie pattern discovery. Proceedings of the Language Resources Evaluation Conference (LREC) (pp. 1742–1745).

Orasan, C. (2003). An evolutionary approach for improving the quality of automatic summaries. Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering (Vol. 12, p. 45). Association for Computational Linguistics.

Pan, J.-Y., Yang, H.-J., Duygulu, P., & Faloutsos, C. (2004). Automatic image captioning. Multimedia and Expo, 2004. ICME'04, pp. 1987–1990.

Panofsky, E. (1972). Studies in iconology: Humanistic themes in the art of the renaissance. New York: Harper & Row.

Purves, R., Edwardes, A., & Sanderson, M. (2008). Describing the where–improving image annotation and search through geography. First International Workshop on Metadata Mining for Image Understanding, Funchal, Madeira-Portugal.

Radev, D., Jing, H., Sty's, M. & Tam, D. (2004), Centroid-based summarization of multiple documents, in 'Information Processing and Management', Vol. 40, Elsevier, pp. 919–938.

Riedhammer, K., Gillick, D., Favre, B., & Hakkani-Tur, D. (2008). Packing the meeting summarization knapsack. Brisbane, Australia: INTERSPEECH.

Rosch, E. (1999). "Principles of categorization", Concepts: Core readings (pp. 189–206). Cambridge, MA: The MIT press.

Saggion, H. (2005). Topic-based Summarization at DUC 2005. Document Understanding Conference (DUC).

Saggion, H., & Gaizauskas, R. (2004). Multi-document summarization by cluster/profile relevance and redundancy removal. Document Understanding Conference (DUC).

Salton, G. & Buckley, C. (1988), Term-weighting approaches in automatic text retrieval, in 'Information Processing and Management: an International Journal', Vol. 24, Pergamon Press, Inc. Tarrytown, NY, USA, pp. 513–523.

Sauper, C., & Barzilay, R. (2009). Automatically generating wikipedia articles: A structure-aware approach. Proceedings of the Joint Conference of the 47th annual meeting of the ACL and the Fourth International Joint Conference on Natural Language Processing of the AFNLP (Vol. 1. pp. 208–216). Association for Computational Linguistics.

Sekine, S. (2006). On-demand information extraction. Proceedings of Association for Computational Linguistics (pp. 731–738), Sydney, Australia.

Shatford, S. (1986). Analyzing the subject of a picture: A theoretical approach. Cataloging and Classification Quarterly, 6(3), 39–61.

Song, F., & Croft, W. (1999). A general language model for information retrieval. Proceedings of the Eighth International Conference on Information and Knowledge Management (pp. 316–321). New York, NY: ACM.

Soricut, R., & Marcu, D. (2006). Discourse generation using utility-trained coherence models. Proceedings of the COLING/ACL on Main Conference Poster Sessions (pp. 803–810). Association for Computational Linguistics.

Stevenson, M., & Greenwood, M. (2005). A semantic approach to IE pattern induction. Proceedings of the 43rd annual meeting on Association for Computational Linguistics (pp. 379–386). Morristown, NJ: Association for Computational Linguistics.

Stevenson, M., & Greenwood, M. (2009). Dependency pattern models for information extraction. Research on Language and Computation, 7(1), 13–39.

Sudo, K., Sekine, S., & Grishman, R. (2001). Automatic pattern acquisition for Japanese information extraction. Proceedings of the First International Conference on Human Language Technology Research (pp. 1–7). Association for Computational Linguistics.

Sudo, K., Sekine, S., & Grishman, R. (2003). An improved extraction pattern representation model for automatic ie pattern acquisition. Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (Vol. 1, pp. 224–231). Association for Computational Linguistics.

Teufel, S. (2010). The structure of scientific articles: Applications to citation indexing and summarization. Stanford, CA: C S L I Publications/ Center for the Study of Language & Information.

Teufel, S., & Moens, M. (1997). Sentence extraction as a classification task. Proceedings of the Association for Computational Linguistics (Vol. 97, pp. 58–65).

Teufel, S., & Moens, M. (2002). Summarizing scientific articles: Experiments with relevance and rhetorical status. Computational Linguistics, 28(4), Cambridge, MA: MIT Press.

Woodsend, K., & Lapata, M. (2012). Multiple aspect summarization using integer linear programming. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 233–243). Association for Computational Linguistics.

Yangarber, R., Grishman, R., Tapanainen, P., & Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000) (pp. 940–946), Saarbriicken, Germany.

Yao, B., Yang, X., Lin, L., Lee, M., & Zhu, S. (2010). I2t: Image parsing to text description. In Proceedings of the IEEE (Vol. 98, , pp. 1485–1508). IEEE

Ye, S., Qiu, L., Chua, T., & Kan, M. (2005). NUS at DUC 2005: Understanding documents via concept links. Document Understanding Conference (DUC).

Yih, W., Goodman, J., Vanderwende, L., & Suzuki, H. (2007). Multi-document summarization by maximizing informative content-words. Proceedings of IJCAI (Vol. 7, pp. 1777–1782).