Proceedings of the Workshop on Automatic Text Summarization 2011

Collocated with Canadian Conference on Artificial Intelligence

St. John's, Newfoundland and Labrador, Canada

May 24, 2011

Organizing Committee

Anna Kazantseva, University of Ottawa
Alistair Kennedy, University of Ottawa
Guy Lapalme, Université de Montréal
Stan Szpakowicz, University of Ottawa

Program Committee

Sabine Bergler, Concordia University
Aurélien Bossard, Orange Labs
Claire Cardie, Cornell University
Giuseppe Carenini, University of British Columbia
Yllias Chali, University of Lethbridge
John Conroy, IDA / Center for Computing Sciences
Pierre-Étienne Genest, Université de Montréal
Atefeh Farzindar, NLP Technologies
Eduard Hovy, University of Southern California
Diana Inkpen, University of Ottawa
Anna Kazantseva, University of Ottawa
Alistair Kennedy, University of Ottawa
Guy Lapalme, Université de Montréal
Vivi Nastase, HITS gGmbH
Thierry Poibeau, CNRS and École Normale Supérieure
Horacio Saggion, Universitat Pompeu Fabra
Frank Schilder, Thomson Reuters
Judith Schlesinger, IDA / Center for Computing Sciences
Josef Steinberger, EC Joint Research Centre
Stan Szpakowicz, University of Ottawa
Kapil Thadani, Columbia University
Lucy Vanderwende, Microsoft Research
René Witte, Concordia University
Florian Wolf, MergeFlow
Liang Zhou, Thomson Reuters

The Workshop Program

8:30 Coffee

8:55 Opening word

9:00 Invited talk I
Bringing Summarization to End Users: Semantic Assistants for Integrating NLP Web Services and Desktop Clients
René Witte

10:00 PathSum: A Summarization Framework Based on Hierarchical Topics
William Darling and Fei Song

10:30 Morning break

10:50 Deep Learning for Automatic Summary Scoring
Pierre-Etienne Genest, Fabrizio Gotti and Yoshua Bengio

11:20 Semantic Modeling of Multimodal Documents for Abstractive Summarization
Charles Greenbacker, Kathleen McCoy, Sandra Carberry and David McDonald

11:50 Lunch

13:20 Invited talk II
The Role of Automatic Summarization in the Canadian Language Industry
Atefeh Farzindar

14:20 Shallow Semantics for Extractive Summarization Using Connexor Machinese Semantics
Darren Kipp

14:50 Afternoon break

15:10 Toward Extractive Summarization of Multimodal Documents
Peng Wu and Sandra Carberry

15:40 Round table
16:40 Final word

Table of Contents

A Word from the Organizers

Automatic Text Summarization (TS) has been a topic of interest in Natural Language Processing for a long time. Many research groups and companies in Canada actively pursue this topic, yet there has never been in Canada a meeting devoted to TS. This workshop, for the first time, brings together Canadian researchers working on TS; we also warmly welcome contributions and guests from abroad. We hope to make small-scale high-quality meetings of the Canadian TS community a tradition.

TS is sometimes considered an NLP-complete problem, in that it touches upon all important NLP techniques, and some others as well. Recent advances in TS have been impressive, but automatic summaries can still be unfailingly distinguished from man-made ones because of their lower quality. Much remains to be done, both in terms of semantic analysis and capturing the main ideas, and in terms of improving linguistic quality of the summaries.

Summarization is the theme of a very influential annual shared evaluation exercise, the Summarization Track at the Text Analysis Conference (TAC). It is not uncommon to plan TS work around this annual event, regardless of its somewhat limited range – it focuses on summarizing news. Our workshop is a venue for work on TS that does not necessarily fit the TAC format. There are papers on using topic modelling for TS (Darling & Song), automatic evaluation of summaries (Genest et al.), summarization of multimodal documents (Greenbacker et al. and Wu & Carberry) and using semantics for TS (Kipp).

We are excited to have Atefeh Farzindar and René Witté as invited speakers. Atefeh will discuss TS in the Canadian language industry, while René will talk about bringing TS to end users.

With a round-table discussion on text summarization at the end of the day, we expect an interesting and productive workshop.

This workshop would not be possible without the hard work of the program committee members. We thank them all for contributing their time and energy to provide high-quality reviews.

Anna Kazantseva, Alistair Kennedy, Guy Lapalme and Stan Szpakowicz

May 2011

Bringing Summarization to End Users: Semantic Assistants for Integrating NLP Web Services and Desktop Clients (invited talk)

René Witte

Concordia University, Computer Science & Software Engineering
rwitte@cse.concordia.ca

E-mails, memos, web sites, news, research papers, reports, and so on: everybody has too much too read and too little time. For more than a decade now, summarization has been promising to support users in dealing with large amounts of textual content – helping to reduce information overload and thereby reducing its negative impacts on productivity.

While summarization techniques have become more sophisticated in recent years, with multi-document summarization, update summaries, focused and contrastive summaries, none of this progress has materialized so far in the desktop tools and applications deployed by everyday users: tools such as e-mail clients, Web browsers, word processors – the primary interfaces where users need summarization – still do not feature any (or only very limited) NLP support.

This talk investigates the reasons behind this lack of summarization adoption and presents a novel way of bringing NLP to end users via "Semantic Assistants"; a project that aims to provide effective means for the integration of natural language processing services into existing applications, using an open service-oriented architecture for NLP Web services. Integrated into desktop applications, such as word processors, email clients, and software development environments, end users can now receive context-specific support for any task involving human language, including different kinds of automatic summarization. Fully open-source, the Semantic Assistants architecture integrates with the GATE framework for NLP and relies on established standards for service description, composition, and execution, in particular ontology (OWL) service models and standard W3C Web services.

About the Speaker

Dr. René Witte is Assistant Professor at Concordia University in Montréal, Canada, where he established the Semantic Software Lab in 2008. He has been working on semantic technologies and knowledge engineering for more than 10 years. His current research focus is the development of foundations for semantic

software engineering and deployment of productive semantic systems for concrete application scenarios. In particular, he works on topics intersecting the areas of software engineering, natural language processing (NLP) and text mining, as well as semantic desktops, knowledge management, database and information systems, and fuzzy theory. Application areas include building engineering, language engineering, biomedical research, information system engineering, and social science. Dr. Witte holds a Doctorate of Engineering (Dr.-Ing.) and a Diploma in Informatics (Dipl.-Inform. (TH)) from the Faculty of Informatics, Karlsruhe Institute of Technology (KIT), Germany.

The Role of Automatic Summarization in the Canadian Language Industry (invited talk)

Atefeh Farzindar

NLP Technologies Inc.
farzindar@nlptechnologies.ca

Summarization technology has been a popular research domain over the past 25 years. Nearly 1.7 billion people go online and many contribute to the content available on the Internet. As a result, the exponential growth of content creation and distribution around the world is creating new opportunities for the language industry. I will discuss how summarization can play a key role as an innovative core technology in a business model, and how to benefit from this radical innovation (as opposed to incremental innovation) in the Canadian language industry. However, summarization is not a business case by itself but part of an information processing or publishing platform. I will talk about some examples of successful automatic summarization in the Canadian Language Industry such as the legal field and e-publishing.

About the Speaker

Dr. Atefeh Farzindar is the founder of NLP Technologies Inc., a company specializing in Natural Language Processing, automatic summarization and statistical machine translation. Dr. Farzindar received her Ph.D. in Computer Science from the Université de Montréal and Paris-Sorbonne University. She is an adjunct professor at the Department of Computer Science at the Université de Montréal. Mrs. Farzindar has made many contributions to research on the automatic summarization and content management system. As president of NLP Technologies, she has managed multiple collaborative R&D projects with various industry and university partners. She is the chair of the language technologies sector of the Language Industry Association (AILIA). Dr. Farzindar is a board member of the Language Technologies Research Centre (LTRC) and co-chair of the Canadian Conference on Artificial Intelligence 2010 and industry chair for Canadian AI'2011.

PathSum: A Summarization Framework Based on Hierarchical Topics

William M. Darling and Fei Song

School of Computer Science, University of Guelph, 50 Stone Road East,
Guelph, Ontario, N1G 2W1, Canada
{wdarling,fsong}@uoguelph.ca

Abstract. We present *PathSum*, a high-performing hierarchical-topic based single- and multi-document automatic text summarization framework. This approach leverages Bayesian nonparametric methods to model sentences as paths through a tree and create a hierarchy of topics from the input in an unsupervised setting. We describe the generative model used to learn a topic tree based on hierarchical latent *Dirichlet* allocation, and an efficient converging algorithm that matches a built-up summary to the theme reflected in the input. We then illustrate how this method encompasses a framework that is amenable to generic and query-focused summarization, and even document creation. We evaluate our method on DUC and TAC data to compare it to standard multi-document news summarization with encouraging results, and in addition we conduct experiments on legal decision summarization to exemplify the generic ability of our method across different domains.

Keywords: text summarization, topic models, bayesian statistics

1 Introduction

Automatic text summarization (ATS) is in many ways an encompassing sub-field of NLP. Researchers in the area often make use of part-of-speech (POS) tagging, named entity recognition (NER), language modeling, and many other techniques in NLP and machine learning. Despite our plentiful access to these state-of-the-art tools and research, however, most complex ATS approaches rarely surpass the results achieved with simple statistics-based methods grown principally out of 60-year-old ideas of term frequency analysis [13, 7]. Nevertheless, more structured statistical approaches, based on Blei, et al.'s latent *Dirichlet* allocation (LDA) [3], have recently been showing promising results through the use of topic- or content-modeling [9, 8]. These approaches perform ATS by modeling input words as being generated from distinct hidden distributions of words. In [9] and [8], *salient* words are seen as emanating from a different source than either *background* or *document-specific* words.

In this work, we present an even more highly structured statistical model where content words are modeled as being generated from a hierarchical topic structure where the most specific topics are at the bottom level and the most

general topic forms the root. Sentences are modeled as being made up of broad words that describe the input at a very general level, but also from more specific sub-topics that are arranged in a tree. To build the tree, we make use of Bayesian nonparametric methods that allow the tree's structure to be organically generated directly from the input data. Following posterior inference, we use the most traveled paths through the topical tree structure to select salient sentences that both represent the document set and at the same time avoid redundancy. In the following sections, we first discuss the related work, and then describe our summarization framework, *PathSum*. This is followed by a discussion of our experiments and results, and a summary of our conclusions and future work.

2 Related Work

2.1 Term Frequency Based Summarization

Term frequency-based summarization has been studied since the 1950's at IBM [11]. In [13], Nenkova, et al. empirically studied a number of human generated summaries and statistically determined that words that appear with high frequency in an input document also appear with high probability in the related human-generated summary. As a result of their study, Nenkova, et al. developed *SumBasic* which assigns a score to an input sentence based on the unigram probability distribution of the words contained within that sentence. This method, as simple as it is, continues to form the basis for many advanced text summarization systems [7].

2.2 Hierarchical LDA

In [2], Blei, et al. describe *hierarchical* latent *Dirichlet* allocation (hLDA). Unlike the basic LDA model [3], where the learned topics are flat and have no stated relation to each other, hLDA allows one to determine the topics that a corpus is made up of, and how those topics relate to each other hierarchically in a tree. The hLDA tree is learned through a nonparametric Bayesian approach where both the number of topics and the structure of the tree are not set *a priori*, but are determined directly from the data through posterior inference [2].

To build the hLDA topic model, Blei, et al. describe the *nested* Chinese Restaurant Process (nCRP), an extension of the CRP. The CRP is a stochastic process that results in a partition of discrete data. When sampling from the CRP, a customer (datapoint) has a finite probability of either joining an existing table (group), or sitting at (creating) a new one, with a probability proportional to a parameter, γ. The nCRP describes an infinite number of infinite-table Chinese restaurants. All customers begin at the "root" restaurant and are seated according to the CRP. Then, each table contains a card that instructs the customer on which restaurant to visit next. At that restaurant, each visitor is again seated according to the CRP, and this process repeats itself infinitely, describing an infinite tree. This nested structure can be used to describe the hierarchical

organization of topics with the more general topics near the root, and the more specific near the bottom. In *PathSum*, a finite version of the hLDA model will be used to create a hierarchical topic model based on sentences.

2.3 Topic-Based Summarization

In [9], Haghighi and Vanderwende describe several probabilistic generative models that are used to learn the words that describe the content in a set of related documents. In such models, a word can be generated from one of three separate latent distributions over words. These include a background distribution for the purpose of modeling stop-words; document specific distributions for words that are less likely to form part of the entire document set's content; and a semantic distribution that models the "content" words shared across a document set. For summarization, the learned semantic distribution is seen as a modeled distribution of the "important" content in the document set. A summary is then built by extracting sentences such that the summary's flat unigram probability distribution is as close as possible to the content distribution.

While a number of other approaches have also applied topic modeling ideas to text summarization (*i.e.* [8] and [1]), recent work by Celikyilmaz and Hakkani-Tur makes direct use of the hLDA model to perform supervised summarization [5]. All sentences in the input set, along with sentences in the related model summaries, are modeled as paths through an hLDA-like tree. When input sentences share a path with summary sentences, this indicates sentence strength and that sentences along this path are likely to appear in a summary. Each input sentence that shares a path with a summary sentence then receives a score based on its similarity to the summary sentence. A regression model is built using the calculated scores and a number of features including n-grams and word-frequency statistics. Support vector regression is employed to train the model and the test-set sentences are scored according to this function. It is important to note that after the initial training, no additional topic modeling is performed and no learned distributions are made use of.

3 *PathSum* Framework

In this section we describe our *PathSum* summarization framework. It is presented as a *framework* rather than a singular approach because a number of the associated methods and implementation details can vary depending on the purpose of the task (summarization or document generation, for instance) and it provides a generic structure for building more specialized systems. A graphical depiction of the framework is shown in Figure 1.

3.1 Motivation

In most natural language documents, and especially in newswire articles (which have become the canonical *de facto* example for multi-document text summarization), an underlying broad theme can generally be discerned along with a

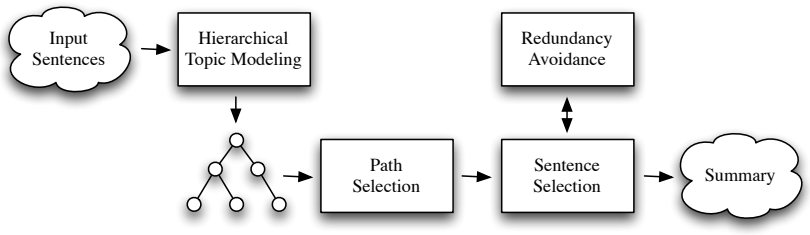

Fig. 1. The *PathSum* summarization framework.

highly varying number of sub-topics, depending on the subject, the author, and the purpose of the document. In traditional statistical summarization, exemplified by methods such as *SumBasic* [13], we can typically do well with respect to the main theme in an input text but there is little possibility to discern between more nuanced sub-topics that are overpowered by popular words that relate to the broad overall topic. This also leads to problems of redundancy as sentences that repeat popular words associated with the main topic are overwhelmingly selected for summary inclusion. By learning the hierarchical topics underlying an input, we could ensure coverage of each sub-story that is expressed in the input. Further, by using a Bayesian nonparametric method that allows the data to determine the model's complexity, we can deal with the varying use of the numbers of sub-themes in input documents.

3.2 hLDA Sentence Modeling

In [2], Blei, et al. use hLDA to model documents as paths through a tree. In our approach, however, so that we can work at a more fine-grained level, and because the sentence unit is typically used in extractive summarization, we model *sentences* as paths $\boldsymbol{\rho}_s$ through a hierarchical topic tree $\mathcal{T}$. Each node in the tree represents a *topic*, or a multinomial distribution over a fixed vocabulary. All sentences share the root topic ϕ_R and then trace a possibly unique path through the tree. In this generative model, a word is generated by sampling a level l from a sentence-specific distribution over levels θ_s. Then, a word is generated from the topic associated with the node at level l along the path $\boldsymbol{\rho}_s$. As with hLDA, the topology of the tree is learned directly from the data. Each sentence path is drawn from the nCRP described above, and as such the branching of the tree emerges from the language used in the input sentences. More formally, sentences in *PathSum* adhere to the following generative process:

1. For each node (topic) $t \in \mathcal{T}$,
   (a) Choose a topic $\phi_t \sim \text{Dirichlet}(\eta)$.
2. For each sentence $s \in \mathbf{S}$,
   (a) Draw $\boldsymbol{\rho}_s \sim \text{nCRP}(\gamma)$.
   (b) Draw $\theta_s \sim \text{Dirichlet}(\alpha)$.
   (c) For each word $w \in s$,

i. Choose level $l \sim \text{Discrete}(\theta_s)$.
ii. Choose word $w \sim \text{Discrete}(\phi_{[\boldsymbol{\rho}_s, l]})$.

Note that unlike the infinite hLDA model described in [2], our initial model represents a fixed-level tree and the level-mixing portions θ_s can therefore be drawn from a finite Dirichlet distribution as opposed to the more complicated GEM distribution required for an infinite depth model.

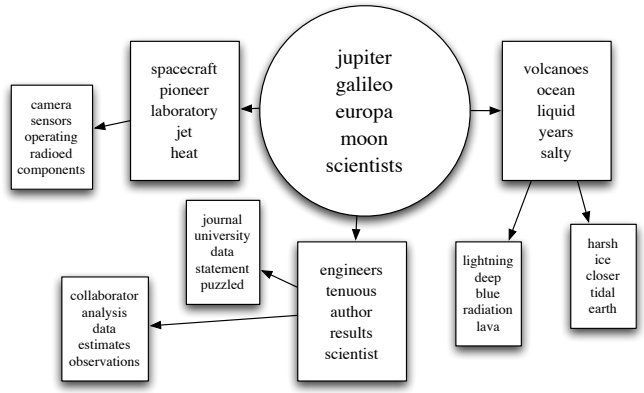

Fig. 2. A 3-level topic tree of the DUC 2006 document set describing NASA's *Galileo* mission to Jupiter's moons. The top 5 words in each topic are shown.

The generative process described above casts sentence generation as a random process where either an existing path can be chosen, or a novel one added to the tree when the data does not fit. Following path selection, a topic is then repeatedly sampled from that path and a word is subsequently sampled from that topic. When we perform posterior inference to learn the path allocations and topic distributions, we can re-create the topic tree. Figure 2 shows a learned tree from the "*Galileo* Mission" document set in the DUC 2006 dataset. The broad topic, presumably shared by all of the sentences in the input, puts high probability on words like *jupiter*, *galileo*, and *moon* that should be generally important in any discourse on that general topic. The next level makes clear distinctions between semantically understandable subtopics that include the astrogeography of Jupiter's moons, the spacecraft itself, and the scientists and engineers involved in the project. Underneath these subtopics lie even more specific *sub-sub*-topics.

3.3 Inference

To learn the posterior distributions $\boldsymbol{\rho}_s$ (the sentence path allocations), ϕ_t (the topic distributions over words), and θ_s (the sentence distributions over levels in the tree), as in [2], and described fully in [6], we appeal to an approximation and use Gibbs sampling. Gibbs sampling is a stochastic approximate inference algorithm from the Markov Chain Monte Carlo (MCMC) family. We set up a

Markov chain with the target posterior as its stationary distribution and take samples from computable conditional distributions that ultimately converge to the true distribution. To converge upon the target more quickly, we use *collapsed* Gibbs sampling where ϕ and θ are integrated out. We sample the sentence path allocations and the per-word level allocations and use these counts to compute the required distributions after a number of iterations.[1]

3.4 Parameter Selection

Despite the name Bayesian *non*-parametric methods, this model still requires parameters to be set *a priori*. However, instead of direct parameters such as the branching factor or the number of topics in the tree, these parameters represent more organic inputs to the model that determine how sparse topics should generally be, and how likely a sentence will be to branch off and create a new topic. We leave more nuanced parameter optimization for future work, but determine our initial settings with the general goal of small trees and interpretable topics. For the topic Dirichlet parameters, we would like to encourage more general topics near the root (when they are viable) and more specific topics near the leaves. Therefore, for a 4-level tree, (unlike the approach outlined in [5]) we set $\eta_1 = 1.0$, $\eta_2 = 0.75$, $\eta_3 = 0.5$, and $\eta_4 = 0.25$, where η_1 represents the root level, etc. For the CRP parameters, we aim to enforce a small number of topics throughout the tree for better generalization, and as such set $\gamma = 0.01$ for each of the three branching levels.

3.5 Extractive Summarization

Each sentence in the input is modeled as a single path from the root to a leaf node in the tree. Because of the clustering nature of the Dirichlet process that produces the topic tree, the themes that are discussed the most in the input – and are therefore seen as the most salient – are represented by the most popular paths through the tree. Therefore, sentences that follow these prevalent paths should be good choices for extraction in building a generic summary of the input. Choosing only the most traveled path would quickly lead to redundancy, however.

Path Selection To produce a strong summary with broad coverage and low redundancy, we take a probabilistic approach and iteratively extract sentences with the goal of matching the input document set's distribution of sentences over paths. If we knew *a priori* how many sentences were going to be extracted, this would be easy. However, the typical task in summarization involves a word limit. We would like a converging approach where as sentences are added, the difference between the input documents' path distribution p and the summary's path distribution q approaches 0, or more formally $\lim_{S_I \to N_S} distDiv(p, q) = 0$,

[1] We used 100 iterations for a burn-in period and then 1000 subsequent iterations to approximate the mode.

where S_I is the set of sentences chosen for extraction, N_S is the number of sentences in the input set, and *distDiv()* measures the difference between the two distributions. Choosing sentences to match the input's path distribution is an important distinction between this approach and *HierSum*, described in [9]. Here, the summary's hierarchical structure is built to match the input's topic hierarchy directly, whereas in *HierSum* the distribution is flattened before the summary is constructed. In that approach, more nuanced (but nevertheless important) topics may therefore be missed by being overpowered by more common words in other topics.

To match the hierarchical path distribution, we select one branch at a time starting at the root and work our way down. We use the function *distDiv()* to determine the difference between each hypothetical new distribution after adding an allocation to it and choose the node that results in the smallest difference. *distDiv()* works by finding the sum of the differences between each dimension in the two discrete distributions. As an example, if p has 4 possible paths with 10 sentences, 25 sentences, 35 sentences, and 30 sentences allocated to each choice, then the distribution is (0.1, 0.25, 0.35, 0.3).[2] The summary path distribution q starts off with no members in any of its 4 possible paths, and we want to come as close as possible to p with each step where we add one member to a single path, but we do not know how many sentences we can get within a word limit. In the first step we add to the third path because then our distribution (0, 0, 1.0, 0) is different by $0.1 + 0.25 + 0.65 + 0.3 = 1.3$ which is the smallest difference between p and any of the possible q's. At the second step, we add to the fourth path because then we have (0, 0, 0.5, 0.5) which is different by $0.1 + 0.25 + 0.15 + 0.2 = 0.7$, so, we get closer. With this measure, once the two distributions are equal the difference is equal to 0, and as we follow this approach step by step we converge towards 0. This process is described more formally in Algorithm 1.

```
Input: path distributions p and q
Output: best path to take
begin
    bestdiv ← ∞
    foreach dimension i in p do
        add one allocation to q[i]
        current ← distDiv(p, q)
        if current < bestdiv then
            bestdiv ← current
            path ← i
        end
    end
    return path
end
```

Algorithm 1: Algorithm to select the next path.

[2] All distributions must, of course, sum to 1.0.

Sentence Selection Once a path is selected as a specific theme to include in the summary, a sentence from that path must be chosen for extraction. We select the sentence with the highest expected posterior probability under the *weighted* distribution of the topics in the given path. These distributions are constructed using the sentence-specific level distributions θ_s. The topic distributions for each node in the given path are linearly combined with that path's average θ_s as a weighting factor ($\sum_d \theta_{s_d} = 1$). This ensures a more fair distribution because some sentences along the path may sample more heavily from the top of the tree, whereas others may be more biased to the leaf nodes, or be more balanced across all levels.

Redundancy Avoidance Despite the broad coverage that *PathSum* promises across themes, there continues to be the possibility of redundancy within paths. Because of the nature of the framework, a number of distinct approaches could be considered and incorporated. In many approaches sentences are scored using a statistical approach and then redundancy is avoided by ensuring that selected sentences' cosine similarity to already included sentences is below a learned threshold. In *SumBasic* and the MMR framework, redundancy avoidance forms part of the sentence scoring function [4]. In this initial exploratory work, we take the latter *SumBasic*-like approach but we reduce the probability of words that show up in selected sentences more gently. When a sentence has been selected for inclusion for a particular path, as in [7], we update the associated probability distribution for words that appear in that sentence as $p'_{path}(w) = \frac{p_{path}(w)}{2.0}$.

3.6 Query-Focused Summarization

The *PathSum* framework is also amenable to query-focused summarization. After building the hierarchical topic structure from the input sentences, a query would then be situated along one of the paths in the tree $\mathcal{T}$. As in the extractive summarization approach described above, sentences would then be chosen for extraction, but only from this path. However, to more faithfully represent the query sentence, the learned level distribution θ_{S_Q}, where S_Q is the query sentence, could be used to weight the topics along that path such that more broad or more specific topics would be highlighted, depending on the query. For example, if the query's level distribution $\theta_{S_Q} = (0.1, 0.1, 0.1, 0.7)$, then sentences more representative of the highly specific topic at the leaf node of the path would be predominantly chosen for inclusion in the summary.

As an example of query-focused summarization, or more generally "document creation" from a knowledge base, we extracted the *Wikipedia* category "Sports Culture", which consists of 49 pages,[3] and learned the associated *PathSum* hierarchical topic model. We then performed query-focused summarization as described above with the short query "sports memorabilia". We set a soft word limit of 100 words[4] and the output is shown in Figure 3. We note that three of

[3] See `http://en.wikipedia.org/wiki/Category:Sports_culture`.

[4] We define a soft word limit as allowing sentences to be added until the summary's word length is equal to or above the limit due to completing the final sentence.

```
    As years passed and many other sports stars joined their
sports, memorabilia collectors also began to broaden their
horizons. Many items used by famous sports stars or at a famous
event have been sold for many dollars at auctions such as Sothebys
and others. This proficiency has also helped boost the popularity
of sports. The only way to ensure that sports memorabilia is
authentic is to make a purchase from a reputable dealer. In the
1980s, sports cards started to get produced in higher numbers, and
collectors started to keep their cards in better condition as they
became increasingly aware of their potential investment value.
```

Fig. 3. Output of *PathSum* document creation on *Wikipedia* category "Sports Culture" with query "sports memorabilia".

the extracted sentences come from the *Wikipedia* page "Sports Memorabilia". However, the generalizing nature of the topic tree allowed a further sentence (the final one) to come from the related page "Sports card". Our document generation research is at a preliminary stage and we do not provide quantifiable results here. Instead, it is our intention to demonstrate the initial possibilities of the *PathSum* framework and its applicability to other areas of NLP research.

4 Experimental Results and Discussion

To test the capabilities of the *PathSum* summarization framework, we describe formal experiments for extractive news summarization and an exploration of "generic" legal decision summarization.

4.1 Extractive News Summarization

The Text Analysis Conference (TAC) and its predecessor the Document Understanding Conference (DUC) hold annual conferences and adjoining competitions to encourage research in automatic multi-document summarization of news articles. Due to the extremely useful aggregation of data and reference summaries that are provided by these conferences, the associated datasets have become *de facto* standards in the ATS literature. We present results on the DUC 2006 dataset due to its popularity, and on the more recent TAC 2010 dataset. The former consists of 50 sets of 25 news articles each and summaries may be a maximum of 250 words. TAC 2010 data is made up of 46 sets of 10 news articles each where the summary word limit is 100 (we only make use of the *initial* TAC 2010 data).

We report our results using the n-gram matching ATS metric ROUGE [10]. We make use of R-1 (unigram), R-2 (bigram), and R-SU4 (skip-4 bigram) both with and without stopwords removed from the calculation. *PathSum* is compared to a baseline where the first sentences of each document in the input set are extracted up to the maximum word length, and against the venerable statistical summarization system *SumBasic*. Despite lacking news domain-specific features as most TAC systems contain, *PathSum*'s ROUGE scores are statistically no

Method	DUC 2006						TAC 2010					
	ROUGE			ROUGE (-s)			ROUGE			ROUGE (-s)		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
First sens	36.8	6.7	12.2	24.6	4.6	7.1	30.4	6.1	9.6	21.8	5.5	6.6
SumBasic	36.9	6.8	12.0	26.4	5.1	7.6	33.2	7.5	10.8	25.7	6.4	7.7
PathSum	**38.2**	**7.6**	**12.9**	**30.6**	**6.6**	**9.6**	34.4	8.3	**11.4**	**28.2**	**7.7**	**9.1**

Table 1. ROUGE Results for DUC 2006 (left) and TAC 2010 (right). Results statistically significantly better than *SumBasic* are displayed in **bold**; results statistically significantly worse than *SumBasic* are underlined.

different than the second-highest performing system for R-2 and the fifth-highest performing system for R-SU4.

As noted in [9], obtaining statistical significance in ROUGE scores is "quite difficult." Nevertheless, *PathSum* proves to perform very strongly and beats *SumBasic* with statistical significance in all six of the ROUGE measures analyzed for DUC 2006 and in four of the six measures for TAC 2010.[5] None of our results were significantly worse than *SumBasic*. For DUC 2006, *PathSum* beats *SumBasic* by 29% for stopword-removed bigram matching, and by 26 % for stopword-removed skip-4 bigram matching. Further, all three stopword removed ROUGE scores are higher than those reported for the most advanced model described in [9], *HierSum*. Please see Table 1 for full results.[6]

4.2 Legal Decision Summarization

One of the most promising applications of ATS is in summarizing legal decisions for jurists. Summarizing the analysis that a judge undertakes in deciding a case is what many lawyers (and especially law students) spend most of their days involved with. Legal writing is specialized enough that domain-dependent, supervised approaches such as [12] will ultimately be required for results that lawyers can confidently depend on. However, statistical approaches will surely form an important module in more complex legal summarization systems. Due to *PathSum*'s ability to discern between nuanced sub-topics, we felt it could produce a strong legal summary compared to simpler approaches such as *SumBasic*. To verify this, we created summaries of the analysis portion of five random Supreme Court of Canada (SCC) cases from 2008. As in [12], we first used thematic segmentation to divide the cases into *Introduction*, *Facts*, *Legal Analysis*, and *Conclusion*. For this experiment we chose to concentrate on legal analysis and therefore only used that segment as input. As above, we use ROUGE for scoring our summaries and for a model summary, we use the analysis portion of the head-note that accompanies all SCC decisions.[7] Finally, for the maximum

[5] We judge statistical significance using the t-test with 95% confidence.

[6] All reported ROUGE scores in this paper are scaled by 100 to aid in readability.

[7] SCC head-notes are typically structured as a summary of the facts, followed by the holding, followed by a summary of the legal analysis. Our model summaries consist of this final portion.

summary word-length, we use a dynamic number where the limit is equal to the number of words in the model summary. We compare an unmodified version of *PathSum* to *SumBasic* and a baseline similar to that used in [12] where the first *max* words of the analysis section are extracted.

Method	ROUGE			ROUGE (-s)		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
First n	51.7	17.6	25.2	32.7	9.2	12.5
SumBasic	63.0	28.9	34.7	50.4	20.0	22.7
PathSum	65.7	31.5	36.3	**56.8**	**24.1**	**26.5**

Table 2. ROUGE Results for legal summarization on SCC 2008 decisions. Results statistically significantly better than *SumBasic* are displayed in **bold**.

As can be seen in the full results in Table 2, *PathSum* performs very well. It beats *SumBasic* in every test and does so with statistical significance in every stopword-removed ROUGE category we measured. We believe that the advanced results are directly attributable to *PathSum*'s approach of attaining broad coverage by summarizing each sub-theme represented in the input set. For example, one of the randomly selected cases we experimented on was *Apotex v. SanofiSynthelabo*, a patent case.[8] In most patent cases, one party is trying to argue that a patent is invalid. There are a number of tests to determine if a patent should be invalidated, and three of the approaches used in this case were *double patenting*, *obviousness*, and *anticipation*. In observing the learned topic tree (like Figure 2), all three of these were clearly delineated as *sub-sub*-topics. *PathSum* was thus able to extract sentences related to each of these areas whereas *SumBasic* could only concentrate on sentences that contained terms popular throughout the entire document. Although this is only a preliminary exploration of using our framework for legal decision summarization, we are encouraged by these initial results.

5 Conclusions and Future Work

In this paper we have described a fully unsupervised hierarchical-topic based summarization framework built atop the hLDA model. Due to its focus on subtopics within an overall main theme, summarization systems built with this framework should result in wider coverage than similar statistical approaches that take a flat view of the input. We have also detailed a basic implementation of a generic multi-document summarization system that not only outperformed some of the most advanced methods described in the literature for traditional news summarization, but that also performed strongly in the more specialized area of legal decision summarization.

This framework provides a number of avenues for future research. In [5], it is noted that when using a previously learned topic model it can be difficult

[8] *Apotex Inc. v. SanofiSynthelabo Canada Inc.*, [2008] 3 S.C.R. 265, 2008 SCC 61.

or impossible to situate new documents within it. In the present description of *PathSum*, like the approaches in [9, 8], we re-learn the topic model for each set of input sentences. This might not be feasible for a large-scale query-focused document creation method. For this, we will look into *online* inference where the model can be efficiently adjusted on the fly as documents are streamed in. Another area of future work is in hierarchical query modeling. With the simple approach described in section 3.6, the words are modeled along a path of the tree but a user may want to emphasize a certain part of his or her query as the general area, and another part as something more specific within that topic.

References

1. Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL 2004*, pages 113–120, 2004.
2. David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2), 2010.
3. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
4. Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR 1998*, pages 335–336, New York, NY, USA, 1998. ACM.
5. Asli Celikyilmaz and Dilek Hakkani-Tur. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 815–824, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
6. Freddy Chong Tat Chua. Derivation of gibbs sampling equation for hierarchical latent dirichlet allocation. Technical report, Singapore Management University, School of Information Systems, 2010.
7. William M. Darling. Multi-document summarization from first principles. In *Text Analysis Conference*, 2010.
8. Hal Daumé, III and Daniel Marcu. Bayesian query-focused summarization. In *ACL 2006*, pages 305–312, Morristown, NJ, USA, 2006. Association for Computational Linguistics.
9. Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *NAACL 2009*, pages 362–370, Morristown, NJ, USA, 2009. Association for Computational Linguistics.
10. Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proc. of the ACL Workshop on Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
11. H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, 1958.
12. Mehdi Yousfi Monod, Atefeh Farzindar, and Guy Lapalme. Supervised machine learning for summarizing legal documents. In *Canadian Conference on AI*, pages 51–62, 2010.
13. Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *SIGIR 2006*, pages 573–580, New York, NY, USA, 2006. ACM.

Deep Learning for Automatic Summary Scoring

Pierre-Etienne Genest[1], Fabrizio Gotti[1], and Yoshua Bengio[2]

Université de Montréal
Département d'informatique et de recherche opérationnelle
Laboratories RALI[1] and LISA[2]
{genestpe,gottif,bengioy}@iro.umontreal.ca
rali.iro.umontreal.ca, www.iro.umontreal.ca/~lisa

Abstract. Automatic summary scoring is used very often by summarization system developers to test different algorithms and to tune their system. We have developed a new approach based on representation learning, using both unsupervised and supervised learning components, to score a summary based on examples of manually evaluated summaries. Our deep learning approach greatly surpassed ROUGE in terms of correlation with Pyramid (content) scores for individual summaries. However, ROUGE performed slightly better when comparing summarization systems based on their average score.

1 Introduction

Progress in the field of Text Summarization requires ways of assessing the quality of summaries. Comparisons and ranking of summarization systems in international evaluations like the Text Analysis Conference (TAC) [7] rely first on manual summary scoring. On the other hand, automatic summary scoring is used very often by summarization system developers to test different algorithms and to tune their system. Although automatic summary scoring is not as reliable, it does not require human resources nor a long time to complete, so it can be repeated as often as necessary.

The best known and most trusted automatic metric for summary evaluation is the so-called Recall-Oriented Understudy for Gisting Evaluation, or ROUGE [16]. This metric is strictly based on n-gram similarity scores between a model summary and the summary to be evaluated.

In this paper, we describe a new approach to automatic summary scoring based on representation learning, using both unsupervised and supervised learning components, to score a summary based on examples of manually evaluated summaries. This is based on recently introduced algorithms for deep learning of representations [12, 11, 1], and is based on a novel architecture for comparing the learned representations associated with two preprocessed summaries, in the spirit of so-called Siamese Networks [5]. Like ROUGE, it also relies on a comparison between a model summary and the summary to be evaluated. This is accomplished in three steps. First, we preprocess the summaries so that they can be expressed as a vector in term-space. The second step attempts to learn

a mapping from the term-space into a concept-space representation of much smaller dimensionality, using an unsupervised auto-encoder [19, 4, 11] trained on a large corpus. That learned intermediate lower-dimensional representation is more abstract and more oriented towards semantics than the raw term-space representation, because combinations of words that have a similar meaning tend to be represented by nearby vectors in that space, in a way similar to Latent Semantic Analysis [8]. Finally, the third step stacks a regression neural network on top of the attribute-wise comparison obtained from the concept-space representations of the summary to be evaluated and of the model summary.

An important advantage of many Deep Learning algorithms is that they can exploit large quantities of unlabeled data to learn better representations [1], that can generally be more easily transferred across different domains [2] or combined with labeled data for semi-supervised learning [24]. The basic hypothesis explaining these earlier successes [9] is that for the type of tasks at hand (and presumably most tasks considered for AI), representations $h(x)$ of inputs $X = x$ that are useful at characterizing $P(X)$ are useful at characterizing $P(Y|x)$ (where Y's are target labels to predict).

Section 2 will describe existing summary evaluation metrics. We give the details of our approach, including how we implemented it, in section 3. Section 4 discusses our results, and we conclude in section 5.

2 Summary Evaluation Metrics

2.1 Direct Manual Metrics

Direct manual metrics produce human-made scores given by subjective criteria. They are generally scaled by integers between 1 and 5 or 1 and 10. The most common, called Overall Responsiveness in the TAC conferences, answers a question such as "Is this a good summary of the document(s)?" and "How much would you pay for this summary?". The other common measure is a linguistic quality score, which assesses a summary's grammaticality, as well as its focus, coherence, etc.

2.2 Pyramid

The Pyramid metric [17] is an indirect manual evaluation metric of a summary's content. Human assessors read each model summary and determine each one's Semantic Content Units (SCUs) – the ideas or statements of a text. The Pyramid content score of a summary to be evaluated is given by the recall of model SCUs present, weighed by the number of model summaries that contained each SCU, if more than one model was available. The Pyramid score is the one we attempt to predict with our approach.

2.3 ROUGE

ROUGE [16] is an automatic evaluation metric that computes an n-gram similarity score between the model summary and the summary to be evaluated. Several

types of ROUGE measures exist, and the one with the highest correlation with manual scores is ROUGE-2 recall – the recall of model summary bigrams. Very high correlations between manual metrics and ROUGE have been observed [6].

3 Our Approach

This section has five subsections as follows. First, we describe the data sets used for training and testing. Then, we describe the three steps of our approach, namely preprocessing the documents into a binary vector in term-space, learning a representation of this term-space into a concept-space, producing a comparative vector (comparing the attributes of the two summaries), and training a neural network to predict the Pyramid score. These are described at a high level, with the last subsection giving technical details of the implementation.

3.1 Data Sets

Most machine learning approaches require large data sets to perform well, and examples of manually scored summaries are relatively rare. The TAC conferences offer a good source of such data and we used the 2008 and 2009 sets because they were recent, had short (100-word) summaries, and were both done using the same task description, namely query-focused, multi-document, 100-word summaries. Together, they contained roughly 4,000 manually evaluated summaries (excluding update summaries which respond to a different task), completed on 92 document sets. The Pyramid scores of these summaries were normalized to be in the range [0.001, 0.999] for easier use by our algorithm.

The documents for which those summaries were written are NewsWire articles found in the Linguistic Data Consortium's AQUAINT-2 corpus [23], which contains more than 900,000 articles from six different news agencies. The autoencoder is trained using this corpus.

3.2 Preprocessing

It is desirable to represent data by its meaningful features for input to a machine learning algorithm. An easy way to do this for text is to use the so-called bag-of-words approach of representing a text by the terms it contains, regardless of the number of times each term occurs.

In this formalism, each document is represented by a binary vector of size equal to the number of terms in a given vocabulary. Each term in the vocabulary has an index in the vector, which corresponds to a dimension in this vector-space, or term-space. A document is thus a point in term-space, defined as a vector with 1's in dimensions corresponding to terms it contains, and 0's in all other dimensions.

Not all terms are important, however, and it is computationally impracticable to deal with a very large vocabulary, so we had to significantly reduce its size. First, the Porter Stemmer [18] is used on the terms, to represent identically

all terms of the same family. A vocabulary of 850,000 unique stems was initially found in the AQUAINT-2 corpus. All numbers and amounts were projected onto a single tag, and all terms that contain special characters were removed, taking care of a large portion of this number. Stop-words were also taken out, in order to prevent the noise the most common English words would likely create. Finally, we kept only the 10,000 most frequently occurring stems in the AQUAINT-2 corpus, a number high enough to cover terms from a wide range of subjects. This corresponds to terms with a minimum frequency of at least 2,500 within the 900,000 articles of the corpus.

3.3 Deep Learning and Auto-Encoder for Dimensionality Reduction

One of the basic ideas behind Deep Learning algorithms [1] is to exploit unsupervised learning to learn intermediate representations that can then be used in a supervised learning framework. In our work, an auto-encoder [11] is used to reduce the dimensionality of the term-space vectors, so that they can be represented in a much smaller concept-space. Auto-Encoders are a special type of multi-layer neural network with a hidden layer of small dimensionality, which represents an encoding that we are trying to learn. In our case, the encoding can be interpreted as a concept-space, made of learned non-linear transformations of the term-space, which can express the most essential features of a document. The encoding is learned by using the large AQUAINT-2 corpus for input. An important characteristic of such representation-learning algorithms is that they can exploit large quantities of unlabeled data. When they are trained on a bag-of-word, they learn a distributed semantic representation for each word [20], in which semantically similar words (or bags-of-words) are associated to nearby vectors.

The process is illustrated in Figure 1. The encoding function $\mathbf{f}$ is applied to the term-space vector $\mathbf{x}$ representing a corpus document in order to produce the concept-space vector $\mathbf{y}$. $\mathbf{f}$ consists of a linear transformation followed by a non-linear function. Next, function $\mathbf{g}$ decodes $\mathbf{y}$ back to a vector in term-space by applying again a linear transformation followed by a non-linear function. The auto-encoder is trained with the goal of learning the parameters of functions $\mathbf{f}$ and $\mathbf{g}$, so that the reconstructed vector $\tilde{\mathbf{x}}$ is similar to the original vector $\mathbf{x}$.

To assess the quality of the encoding and decoding, a loss function $\mathbf{L}$ matched to the non-linearity compares $\mathbf{x}$ and $\tilde{\mathbf{x}}$. The parameters of the auto-encoder are set to minimize the reconstruction error on the training data.

Because the input vectors are binary and sparse (95% of the dimensions contain zeros), we apply a sampling algorithm to speed up the processing. All dimensions that contain ones are sampled, as well as the same number of dimensions containing zeros, selected randomly. Reconstruction and error gradients are only computed for the sampled dimensions, i.e., no gradient is back-propagated for the non-sampled dimensions. To our knowledge, this sampling mechanism is novel in the context of auto-encoders, and it allowed us considerable speed-up of training time, around 8-fold. This is a major advantage to explore such un-

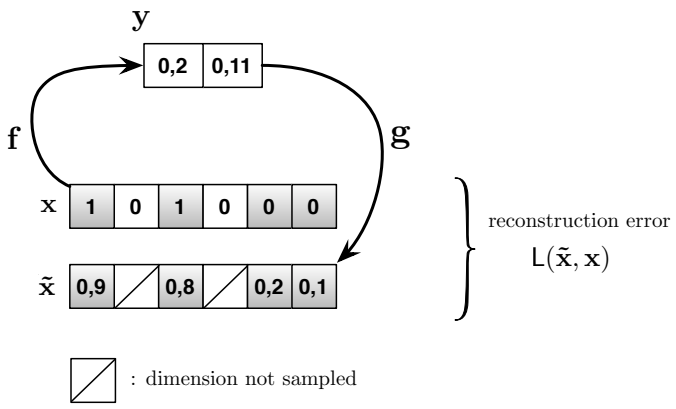

Fig. 1. Architecture of the auto-encoder to learn an encoding **y** in concept-space of an input **x** in term-space.

supervised learning algorithms in the context of the large number of training examples used (around a million).

Four values for the number of dimensions of the concept-space were tested: 200, 400, 600 and 1,000. A lower average reconstruction error for the auto-encoder, as well as a better performance when this encoding is used as part of the regression process were observed when the input is encoded in a concept-space of 600 dimensions.

Although the reconstruction error criterion of an auto-encoder does not correspond to training a probabilistic model of the input vectors, a slight modification of it, called the denoising auto-encoder [22], and in which the auto-encoder takes a corrupted input and tries to reconstruct the original clean input, does. A variant of the denoising auto-encoder corresponds [21] to applying a regularized Score Matching criterion [13] to a particular Restricted Boltzmann Machine [11, 12]. In our experiments we found that the addition of this corruption process helped, but only marginally, so results with the simpler ordinary auto-encoder are reported here.

3.4 Supervised Regression on Top of Learned Representation

The objective is to compare a summary **s** with a model summary **m**. On top of the representation learned by the auto-encoder for the summary **s** and for the model summary **m**, we first compute a "comparison" layer that performs an element-wise comparison between the two summaries' concept-space attributes, on top of which we then stack a supervised multi-layer regression neural network, to predict the summary Pyramid score p. The layout of the network is illustrated in Figure 2.

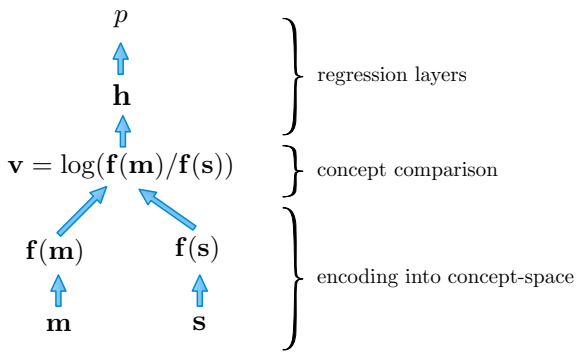

Fig. 2. Deep Learning architecture for regression, with three steps: encoding into concept-space, concept comparison and regression layers.

The input layer contains the two preprocessed summaries to be compared, represented by term-space vectors. The encoding function $\mathbf{f}$ learned by the auto-encoder is applied on both term-space vectors to reduce them to their concept-space representation.

The layer $\mathbf{v}$ compares the values of $\mathbf{f}(\mathbf{s})$ and $\mathbf{f}(\mathbf{m})$ for each concept (how strongly each concept is present in the summary and the model). For each dimension in concept-space, $\mathbf{v}$ is computed with the element-wise logarithm of the ratio of $\mathbf{f}(\mathbf{s})$ and $\mathbf{f}(\mathbf{m})$.

The remainder of the architecture is a standard one-hidden layer neural network for regression. The size of the hidden layer $\mathbf{h}$ was set to 1,000, because a much higher number of hidden units would have entailed a large increase in computing time yielding meager gains in performance.

Architecture variants. Four variants of the architecture have been considered, in order to test two hypotheses. Firstly, the impact of the auto-encoder unsupervised pre-training was tested. The function $\mathbf{f}$ of the architecture would either be initialized randomly, or by the learned function of the auto-encoder in order to test this hypothesis. Secondly, we tested whether or not the parameters of the concept-space encoding function $\mathbf{f}$ should be adapted during training or left untouched. Adjusting the parameters of $\mathbf{f}$ during training severely slows down the execution. Our experiments showed that there is always a very significant gain to use the auto-encoder-learned concepts rather than random ones, even when those can be adjusted during training. Also, statistical tests failed to observe a significant difference between keeping the learned concepts fixed and adjusting them to new data during training. These findings are different from those of previous work using auto-encoders in deep architectures [3, 14], which observed better results when all the parameters could be adjusted to the data.

3.5 Technical Details about the Implementation

Hyper-parameters For both the auto-encoder and the regression component, we tested several values for each of the hyperparameters of the algorithms, using a grid search. These include the size of the hidden layers and variations of the algorithms themselves, as already mentioned, but also the learning rate for gradient descent, its rate of decay and the number of iterations for early stopping. The gradient descent learning rate was taken as $\epsilon_t = \frac{\epsilon_0 \tau}{t+\tau}$, where ϵ_0 is the initial learning rate and τ controls the rate of decay (asymptotically in $1/t$, to guarantee convergence).

Auto-Encoder The two layers of the network are computed using $\mathbf{y} = \mathbf{f}(\mathbf{x}) = \tanh(\mathbf{W}\mathbf{x} + \mathbf{b})$ and $\tilde{\mathbf{x}} = \mathbf{g}(\mathbf{y}) = \sigma(\mathbf{W}'\mathbf{y} + \mathbf{b}')$, where σ is the logistic sigmoid. Therefore, $\mathbf{f}$ is defined by a linear transformation matrix $\mathbf{W}$ and two bias vectors $\mathbf{b}$ and $\mathbf{b}'$. We used tied weight matrices, so that $\mathbf{W}' = \mathbf{W}^T$. The loss function used is a cross-entropy between the reconstruction and the original vector, given by $\mathsf{L}(\tilde{\mathbf{x}}, \mathbf{x}) = -mean(\mathbf{x}\log(\tilde{\mathbf{x}}) + (1-\mathbf{x})\log(1-\tilde{\mathbf{x}}))$ where the mean is taken over the elements of $\mathbf{x}$. The matrix $\mathbf{W}$ is initialized randomly and uniformly, such that $W_{ij} \in [-\frac{1}{\sqrt{\bar{u}}}, \frac{1}{\sqrt{\bar{u}}}]$, with $\bar{u}$ the average number of ones in $\mathbf{x}$. The bias vectors were initialized with $\mathbf{b} \in [0, 2]$ and $\mathbf{b}' \in [-0.5, 0.5]$. These values were chosen to stay close to inflection points of the non-linear activation functions of each layer, as suggested in LeCun [15].

Comparison Layer and Regression Component The equation used for the comparison layer is slightly different from that in Figure 2, to avoid taking the logarithm of zero:

$$\mathbf{v} = \log\left(\frac{\mathbf{f}(\mathbf{m}) + 1 + \epsilon}{\mathbf{f}(\mathbf{s}) + 1 + \epsilon}\right), \tag{1}$$

where ϵ is 10^{-3}. The regression layers are computed similarly to $\mathbf{f}$ and $\mathbf{g}$. The hidden units $\mathbf{h}$ are computed by taking a linear transformation of $\mathbf{v}$ and applying a hyperbolic tangent. The score p was computed by a linear transformation of $\mathbf{h}$ and applying a logistic sigmoid activation function. The loss function is the Kullback-Liebler divergence between the predicted score and the target Pyramid score. The initialization of each matrix element for all layers was in the uniform interval $\left[-\sqrt{\frac{6}{n_i+n_j}}, \sqrt{\frac{6}{n_i+n_j}}\right]$, with n_i and n_j the number of units of the layer below and the current layer. All biases were initialized to 0.

4 Results and Analysis

4.1 Algorithmic Performance

Auto-Encoder The encoder was trained on 800,000 documents randomly selected from the 900,000 articles contained in the ACQUAINT-2 corpus. The

Correlation with Pyramid scores			
	Pearson	Spearman	Kendall
Deep Learner	0.786	0.782	0.591
ROUGE-2	0.617	0.591	0.427

Table 1. Correlation coefficients between our Deep Learning approach and Pyramid, and between ROUGE-2 and Pyramid ($n = 2288$), for individual summary predictions (comparing a single pair of summaries).

remaining 100,000 documents were set aside for testing purposes. The reconstruction error achieved by the auto-encoder on this test set is a cross-entropy of 0.134. This could for example be achieved by predicting (on average) a value of 0.13 for the dimensions of terms that did not appear in the input document, and predicting 0.87 for the dimensions of terms that did appear in it.

Regressor An 11-fold cross-validation was performed to evaluate the quality of the regression. A single fold consists of a training set including all of the 2008 TAC data and $10/11^{\text{th}}$ of the 2009 data. The remaining $1/11^{\text{th}}$ of the 2009 data constitutes the test set.

The mean absolute error between the predicted score for a summary and the actual Pyramid score in the test set is 0.085. Note that, although the scores are between 0 and 1, most of them are fairly low, with a mean of 0.25 and a standard deviation of 0.17, as can be observed in Figure 3 below.

4.2 Intrinsic Evaluation

In the language of Galliers and Spärck Jones [10], an intrinsic evaluation is related to a system's objective, whereas extrinsic evaluation is interested with a system's actual function. Because our objective is to simulate the Pyramid scores for evaluating summaries, it is interesting to compute correlation coefficients between our predictions and Pyramid scores of the test set. These correlation coefficients serve here as an intrinsic evaluation method. Table 1 shows our correlation with Pyramid and compares it with ROUGE's. Figure 3 plots all the data points for our predictions and ROUGE's.

These results are very satisfactory, as they show that our approach tends to rank summaries according to their content much better than ROUGE does. Although the correlation coefficients for individual comparisons are not very high compared to the correlation coefficients for a whole system (Table 2), they show a decent level of performance for an automatic evaluation metric. More importantly, there is a surprisingly large improvement in individual summary assessment when compared to ROUGE, a 32% relative improvement in Spearman correlation, going from .591 to .782 (see Table 1).

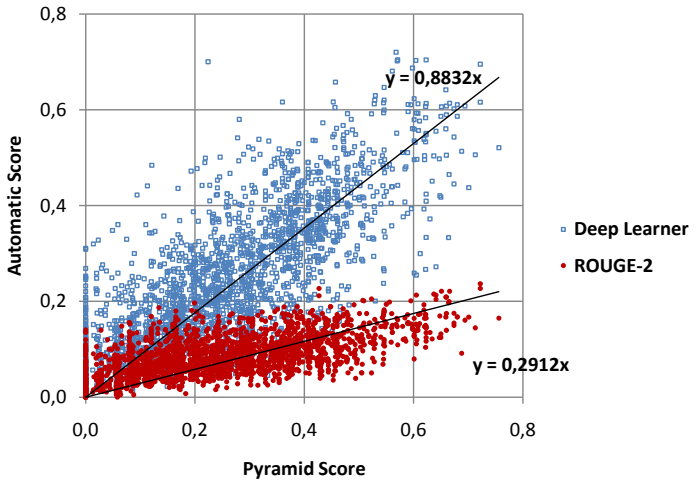

Fig. 3. Plot of the predictions from our Deep Learning approach and ROUGE-2 against Pyramid scores, for each of the 2288 summaries evaluated for testing. A large proportion of points have a Pyramid score of 0.

Correlation coefficients between system averages			
	Pearson	Spearman	Kendall
Deep Learner	0.946	0.898	0.751
ROUGE-2	0.972	0.942	0.803

Table 2. Correlation coefficients between per-system averages ($n = 52$).

4.3 Extrinsic Evaluation

In practice, evaluation metrics are not used to rank single summaries, but systems, or different configurations of a system. This is where automatic metrics like ROUGE are most useful: they allow the fine-tuning of a system, possibly iteratively, without having to manually evaluate the many summaries produced, or to even read any of them. The 2009 TAC data comprised summaries written by 52 automatic systems, so we averaged the scores for each system, over the 44 document sets. This was done for both our approach, Pyramid and ROUGE.

As table 2 shows, ROUGE averages are more correlated to Pyramid averages than our approach's averages are. That is, given many examples of two summarization systems's output, ROUGE-2 predicts slightly more accurately which one is better on average than our approach does. Figure 4 shows the actual data points for each system.

Moreover, this extrinsic discriminative power can be evaluated directly using Welch t-tests between pairs of systems. This is a way to verify how our approach and ROUGE rank any pair of systems, as compared to how Pyramid ranks them,

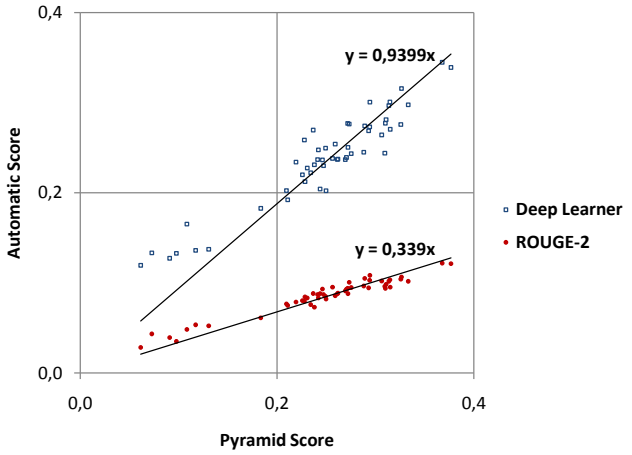

Fig. 4. Plot of averages of the predictions from our Deep Learning approach and ROUGE-2 against Pyramid scores, for each of the 52 evaluated systems.

Discriminative power between systems			
	Agreements	Disagreements	Contradictions
Deep Learner	1107	219	0
ROUGE-2	1124	202	0

Table 3. Number of agreements, disagreements and contradictions between an approach and Pyramid, from Welch t-tests conducted over all possible pairs of summarization systems.

using appropriate statistical tests. From two distributions A and B of predicted scores on two summarization systems, we verify if we observe a statistically significant difference between them using a Welch t-test. The same is done with the distribution of Pyramid scores. An agreement is when both tests produce the same result, namely that A and B are indistinguishable, that A is greater than B or the opposite. A disagreement is when one test shows no significant distinction while the other believes there is one. Finally, a contradiction occurs when one test shows that A is better than B and the other shows the opposite.

Ultimately, it is difficult to understand why our system has better intrinsic performance than ROUGE but slightly inferior extrinsic performance. There could have been some sort of noise-canceling effect for ROUGE that is less present with our approach.

5 Conclusion

We have introduced a way to speed-up training of unsupervised auto-encoders that learn a semantically beneficial representation of concepts from the given

term-space input, based on a sampling approximation of the training criterion. We have shown that an implementation of Deep Learning regression for summary evaluation scores can substantially improve on ROUGE at determining which of two summaries is better, yielding a 32% relative improvement in Spearman correlation, going from .591 to .782. On the task of determining which of two systems are better, and using only a simple averaging aggregation, performance was slightly worse than ROUGE-2 recall. This slightly inferior performance on per-system averages might come from the overspecialization of using a machine learning approach. The (intrinsic) goal that we set out for our approach was to predict Pyramid scores as best as possible for single observations. No specific learning of how to score systems given many example outputs was made, even though it is what most summarization developers actually need. With better assessment at the individual summary level, we believe that there is a great potential to extend these results to the system-level by better aggregation and training methods focused on improving the system-level predicted performance. For this purpose, it would be interesting to explore ways to design a machine learning algorithm which takes a set of input summaries and outputs something better than averaging the score predictions. This could be done by adding an extra layer to the architecture, or by running another algorithm to learn to combine scores efficiently for ranking purposes.

Another variant which should be explored is the insertion of an absolute value computation after the logarithm in equation 1. This would guarantee that the regression is invariant to switching the order of its two input summaries m and s, and $\mathbf{v}$ would measure the absolute discrepancy between various aspects of the summaries captured by each of the dimensions of $f(m)$ and $f(s)$.

References

1. Y. Bengio. Learning deep architectures for AI. *Foundations & Trends in Mach. Learn.*, 2(1):1–127, 2009.
2. Yoshua Bengio, Frederic Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, Xavier Glorot, Xavier Muller, Sylvain Pannetier-Lebeuf, Razvan Pascanu, François Savard, and Guillaume Sicard. Deep self-taught learning for handwritten character recognition. NIPS*2010 Deep Learning and Unsupervised Feature Learning Workshop, 2010.
3. Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In Bernhard Schölkopf, John Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems 19 (NIPS'06)*, pages 153–160. MIT Press, 2007.
4. Herv Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59:291–294, 1988.
5. J. Bromley, J. Benz, L. Bottou, I. Guyon, L. Jackel, Y. LeCun, C. Moore, E. Sackinger, and R. Shah. Signature verification using a siamese time delay neural network. In *Advances in Pattern Recognition Systems using Neural Network Technologies*, pages 669–687. World Scientific, Singapore, 1993.

6. Hoa Trang Dang and Karolina Owczarzak. Overview of the TAC 2009 Summarization Track. In *Proceedings of the Second Text Analysis Conference*, Gaithersburg, Maryland, USA, 2009. National Institute of Standards and Technology.
7. Hoa Trang Dang and Karolina Owczarzak. Overview of the TAC 2010 Summarization Track. In *Proceedings of the Third Text Analysis Conference*, Gaithersburg, Maryland, USA, 2010. National Institute of Standards and Technology.
8. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
9. Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660, February 2010.
10. J.R. Galliers and K. Spärck Jones. Evaluating Natural Language Processing Systems. Technical Report UCAM-CL-TR-291, U. of Cambridge, Computer Laboratory, 1993.
11. G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313:504–507, July 2006.
12. Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.
13. Aapo Hyvärinen. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.
14. Pascal Lamblin and Yoshua Bengio. Important gains from supervised fine-tuning of deep architectures on large labeled sets. NIPS*2010 Deep Learning and Unsupervised Feature Learning Workshop, 2010.
15. Y. LeCun, L. Bottou, G. Orr, and K. Muller. Efficient backprop. In G. Orr and Muller K., editors, *Neural Networks: Tricks of the trade*. Springer, 1998.
16. Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the ACL-04 Workshop: Text Summarization Branches Out*, pages 74–81, 2004.
17. Rebecca J. Passonneau. Pyramid Annotation Guide: DUC 2006. www1.cs.columbia.edu/∼becky/DUC2006/2006-pyramid-guidelines.html, 2006.
18. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
19. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
20. Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics(ACL2010)*, pages 384–394. Association for Computational Linguistics, July 2010.
21. Pascal Vincent. A connection between score matching and denoising autoencoders. Technical Report 1358, Université de Montréal, DIRO, November 2010.
22. Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, New York, NY, USA, 2008. ACM.
23. Ellen Vorhees and David Graff. AQUAINT-2 Information-Retrieval Text Research Collection. Linguistic Data Consortium, Philadelphia, 2008.
24. Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)*, pages 1168–1175. ACM, 2008.

Semantic Modeling of Multimodal Documents for Abstractive Summarization

Charles F. Greenbacker[1], Kathleen F. McCoy[1], Sandra Carberry[1], and David D. McDonald[2]

[1] Department of Computer and Information Sciences, University of Delaware, Newark, Delaware, USA* {charlieg,mccoy,carberry}@cis.udel.edu
[2] SIFT, LLC, Boston, Massachusetts, USA dmcdonald@sift.info

Abstract. We describe a method for semantic modeling of multimodal documents and discuss how this can be used to generate an abstractive summary. Information extracted from the text by a semantic parser and from the graphics by a graph understanding system is combined into a single knowledge base. By operating at the semantic (rather than the surface) level, we are able to integrate information collected from both text and non-text sources. From this unified semantic model, we can evaluate the importance of each part of the extracted knowledge and produce a comprehensive summary of the entire multimodal document.

1 Introduction

This work is part of a larger ongoing effort to produce better and more inclusive descriptions of the information contained in multimodal documents found in popular media. Multimodal documents consisting of text and information graphics (such as bar charts and line graphs) pose a difficult challenge for traditional natural language processing techniques. The graphical content is not always duplicated in the text of the document [4], and yet the graphic may contain valuable information important to the article's message. The content creator had a reason for including the graphic in the multimodal document, and if the graphic is ignored, the summary may not be a good representation of the document as a whole. Our current focus is on combining information extracted from the text with the most important information conveyed by the graphics in order to produce an integrated summary of the entire article. This line of work helps address what is commonly known as the *information overload* problem by condensing the information contained in multimodal documents into brief synopses. This is particularly important for people with visual impairments, due to the significant time investment required for them to read lengthy articles, as well as the additional difficulties they face in accessing graphical content. We approach summarization from a generation perspective, thus our goal is to produce a natural language summary as output.

* This work was supported in part by the U.S. Dept. of Education National Institute on Disability and Rehabilitation Research under grant no. H133G080047.

2 Motivation

A summary which considers the information contained in graphical sources should be abstractive in nature. Most summarization tools utilize extractive techniques [19, 20], whereby the most important sentences are extracted from a document and then reassembled to form the summary. However, this approach cannot faithfully retrieve the information stored in graphics since these non-textual modalities offer no sentences for extraction. Some research into summarizing or otherwise representing the content of a graphic has relied solely on captions and other sentences in the article explicitly referring to the graphic in order to summarize it [2, 34]. However, studies have shown that the graphical content is often not repeated in the accompanying article text [4] and captions are often uninformative [14]. Work on summarizing multimodal documents has taken images and text into account to some extent, by doing very shallow processing on an image to categorize it [11], or using the accompanying text to disambiguate image contents [31], but none that we are aware of consider a graphic on par with text in terms of adding communicative content to a document. Furthermore, summaries produced by extractive methods in general, while syntactically correct, have been shown to lack cohesion and suffer from ambiguity and referent identification issues [26]. In contrast, an abstractive summary would address both of these issues by working from an underlying semantic representation of the text and graphics, and by using natural language generation techniques for text structure and surface realization to ensure text coherence.

One possible approach to facilitate extractive summarization of multimodal documents would be to first generate a textual description of the graphics [12, 7] and then insert this description into the document text before performing sentence extraction. However, not only would the resulting summary suffer from the limitations inherent to extractive methods described above, it would face additional difficulties because the combined text (machine-generated graphical description inserted into original text of article) would be written by two different authors in two different styles, thus leading to even more coherence issues. Therefore, not only do graphics require an abstractive treatment, information from both text and graphics should be semantically integrated in order to generate a cohesive summary of the entire multimodal document.

Automatic summarization methods that more closely approximate the human process of conceptual integration and regeneration in writing an abstract will likely produce results which are more human-like than that of traditional extraction techniques. However, the automatic abstractive summarization of text has proven to be quite a challenging problem [27], even without considering the incorporation of multimodal sources of information. Efforts directed towards abstraction have included the modification (i.e., editing & rewriting) of extracted sentences [18], as well as using partial semantic analysis with text regeneration and elaboration to produce indicative-informative abstracts from technical information [30]. Some research into "semantic abstraction summarization" has aimed to represent the summarized content as a graphical condensate [17], rather than producing a natural language summary. Our work shares similarities with

the knowledge-based text condensation model of Reimer & Hahn [29], as well as with Rau et al. [28], who developed an information extraction approach for conceptual information summarization, though our goal is to represent both the text and the graphics in a single conceptual model in order to generate a natural language summary of a multimodal document.

3 Methodology

In the remainder of this paper, we will present our method for extracting information from text and graphical sources to build a semantic model that captures the information content of both the text and the graphics, and then discuss how an abstractive summary can be produced from this model.

3.1 Text Understanding

The semantic parsing of document text is performed by Sparser [21], a bottom-up, phrase-structure-based chart parser, optimized for semantic grammars and partial parsing.[3] While most parsers stop at a structural description, Sparser produces a disambiguated conceptual model. It outputs categorized objects and relationships, creating and adding specific facts to instances of highly-structured, predefined prototypes. Sparser contains a built-in, sophisticated linguistic model of core English grammar, as well as a model of common items such as names, locations, times, and amounts. Given a document and domain-specific grammar, Sparser performs a linguistic analysis, identifying each part of the text where the subjects of its grammar appear, and emitting partially-saturated referents (PSRs) as a semantic representation of what it recognizes [23]. A PSR is a semantically-incomplete representation of a concept for which some of the characteristic information can be missing; in other words, an object possibly lacking values for some of its attributes.

Existing Sparser grammars provide coverage for several different domains, including business news articles. A collection of multimodal documents from popular media has been assembled, most of which contain article text accompanied by information graphics. Among these articles are many in the business news domain. We have extended Sparser's semantic grammar for this domain, allowing it to analyze texts like the article entitled "Will Medtronic's Pulse Quicken?" from the May 29, 2006 edition of Businessweek magazine.[4] Such texts convey information about stock prices, earnings forecasts, analysts' predictions, and market conditions. Sparser recognizes these semantic entities, builds and modifies PSRs to represent them, and adds these to the semantic model being constructed.

3.2 Graph Understanding

As image understanding research has not yet developed tools capable of extracting semantic content from every possible image, we must restrict our focus to a

[3] `https://github.com/charlieg/Sparser`
[4] `http://www.businessweek.com/magazine/content/06_22/b3986120.htm`

limited class of images for the prototype system implementation. We have opted to leverage capabilities developed for the SIGHT system [9], which generates textual summaries of information graphics found in popular media (e.g., magazines, newspapers) for people who have visual impairments. Rather than focusing on specific data points or the shape of the graphic (as might be appropriate for a scientific graph), SIGHT conveys the underlying message (made apparent by the choice of graph type and the communicative signals entered into the graphic by the graph's author) along with propositions that are highlighted by visual features. For example, given the bar chart in Fig. 1, SIGHT generates the following initial summary [8] in about one minute on a modern PC:

> *Following a moderate rise between the year 1993 and the year 1994, the graphic shows a decreasing trend in the amount of newark rainfall for july over the period from the year 1994 to the year 2002. The amount of newark rainfall for july shows the largest drop of about 1.29 inches between the year 1999 and the year 2000. With the exception of a few rises, slight decreases are observed almost every year over the period from the year 1994 to the year 2002.*

Our framework is general enough to accomodate arbitrary image types and other modalities (e.g., audio, video), however. Incorporating other modalities would require adding a module capable of mapping the particular modality to its underlying message-level semantic content.

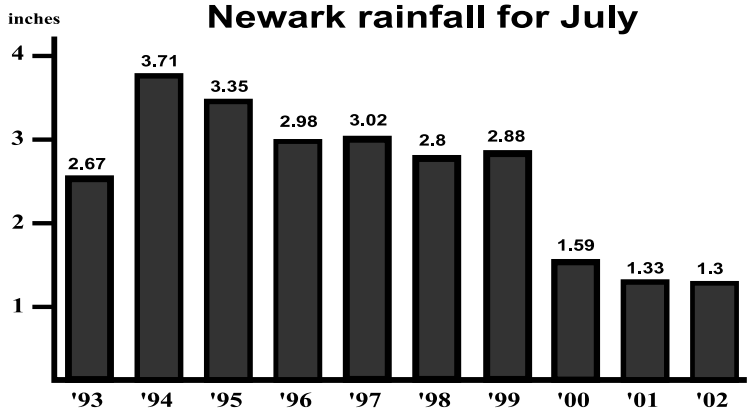

Fig. 1. Example bar chart processed by SIGHT.

Several modules of the SIGHT system are relevant to our current work. The image file is first analyzed by SIGHT's visual extraction module [6], which produces an XML representation of the information stored in the graphic [16]. For example, given a bar chart, the XML output contains axis labels, information about each bar (e.g., position, height, value, color/shading), captions and legends, etc. We contend that this "raw information" extracted from the graphic

(the *visual* level) is not the proper level of understanding upon which to base a summary of the article. Far more pertinent is the communicative intent of the graphic as it relates to the overall document (the *message* level). Thus, SIGHT's intention recognition module [15] applies an inference model (including Bayesian networks) to reason about the communicative signals contained in the graphic (based on attributes derived from statistical tests, cognitive psychology research into perceptual task effort, and visual features) to identify the intended message of a bar chart (e.g., rising trend, rank of an entity). Recent work has extended this to line graphs [32, 33] and a subclass of grouped bar charts [3].

Once the communicative intent has been identified, the system extracts additional salient propositions that expand on the graph's intended message. On the basis of human subjects experiments, the propositions are marked with varying levels of importance depending on the intended message and visual features of the graph. These propositions, along with the intended message, represent the knowledge conveyed by the graphic and capture the *message-level* content that the graph should contribute to the summary of the document. The propositions capture a variety of concepts, including time span, degree of volatility, exceptions in trends, and entity comparisons. The inferred message and extracted propositions are added to the semantic model, making connections to concepts previously derived from the text as appropriate. The SIGHT system is already capable of extracting the most salient propositions from simple bar charts [7], and current efforts are working to extend this capability to line graphs and grouped bar charts as well.

3.3 Knowledge Representation

For our knowledge representation system, we use KRISP ("Knowledge Representation In SParser"): a system of typed, structured objects organized under a foundational ontology [22]. The PSRs recognized by Sparser are stored in KRISP as instantiations of pre-defined categories in a model. As Sparser obtains more and more information about a particular object, the corresponding entry in the model becomes more complete (i.e., "filled-out" or "saturated"). In addition, meta-information relating to the concept, such as document structure (e.g., position in the text) and the use of rhetorical devices (e.g., appearance in a comparison by means of juxtaposition), is included in the model as well. Finally, the model stores the original phrasings used in the source document to express each concept in the form of tree-adjoining grammar (TAG) derivation trees, which are the underlying syntactic representation for Sparser; these phrasings are made available for use during the generation phase.

The semantic model of the text built by Sparser is extended to cover the entire multimodal document by decomposing the intended message and propositions extracted from the graphics by SIGHT and inserting this information into the model. Though the graphs often contain material not repeated in the text, there is usually a high degree of connectedness between concepts presented in the text and those in the information graphics. This is represented in the model by instantiating the new objects and relationships introduced by the graphs,

forging new connections to existing entries, and filling the slots of previously-observed PSRs as appropriate. In addition, mirroring the document structure and rhetorical device details associated with the text-based concepts, the propositions extracted from the graphic are marked with importance values derived from the human subjects experiments. These ratings are influenced by the intended message and visual characteristics of the graph.

Sample Semantic Model Figure 2 offers a high-level overview of the semantic model built for the Medtronic article mentioned in Sect. 3.1, while Fig. 3 provides a detailed look at a zoomed-in section of the same model. Each node in Fig. 2 represents an individual concept recognized in the document either by Sparser or the graph understanding component. The name indicates the conceptual category with a number to distinguish between instances. In the interest of space, individual attributes of model entries have been omitted from this diagram, but are available in Fig. 3. Lines connecting nodes indicate a semantic link between the corresponding concepts (i.e., one fills an attribute slot of the other). In addition to entities from the text recognized by Sparser, this diagram also shows the overall intended message (ChangeTrend1) and informational propositions (e.g., Volatile1) the SIGHT analyzer extracted from the graphic. This way, information gathered from text and graphical sources can be integrated at the *semantic* level regardless of the format of the source.

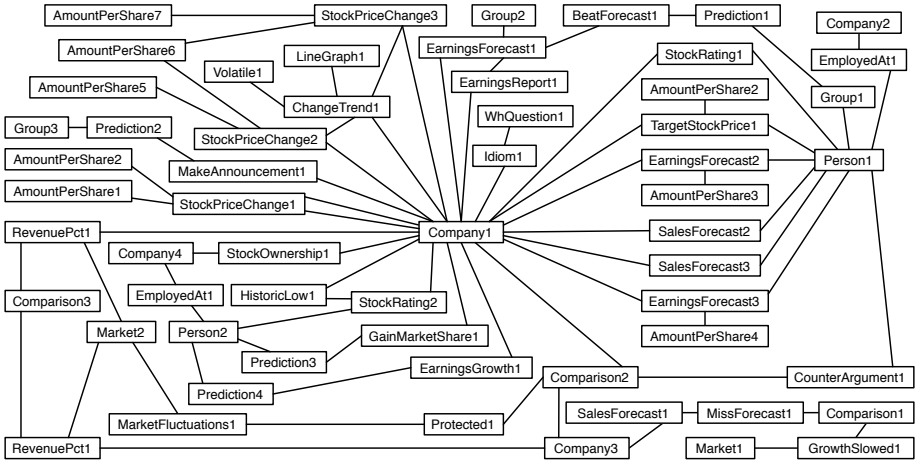

Fig. 2. High-level (low-detail) representation of semantic model for Medtronic article.

Figure 3 zooms into a portion of the model to show more detail for individual concepts. The top section of each box contains the category name and instance number, the middle section shows various attributes with their values (if any), and the bottom section lists the original phrasings expressing these concepts

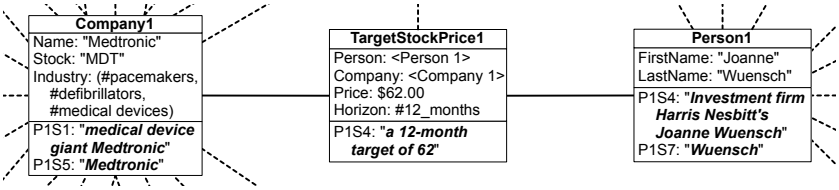

Fig. 3. Low-level detail of model showing concepts used in Sect. 4 sample output.

(formally stored as a synchronous TAG). Attribute values in angle brackets (<>) are references to other concepts. Hash marks (#) denote a symbol that has not been fully instantiated as a concept in the current model.

Encapsulating Document Structure The model also tracks various details regarding document structure. Each recorded expression is marked with a sentence tag (e.g., "P1S4" stands for "paragraph 1, sentence 4" as seen in Fig. 3), indicating exactly where each concept appeared in the text. This allows the content rating metrics to take into account the location of a referent, whether mentioned in the title or buried in the middle of a paragraph, when determining salience. Information obtained from graphical sources receives a similar treatment: entries are marked with importance values derived from our analysis of the corresponding propositions (e.g., due to their rating in our human subjects experiments). As such, the fact that a particular concept is featured prominently in an information graphic is considered during content rating. Certain rhetorical devices that highlight a concept's significance are accounted for as well and represented as distinct entries in the semantic model (e.g., Comparison2 and Idiom1 in Fig. 2). We can accomodate documents of any length, limited only by the storage and processing capacities of the computing environment. Dealing with longer documents is not necessarily more difficult than shorter ones. Articles with a high degree of focus on a central theme tend to result in elaborating and extending existing concepts, rather than introducing new ones. As a result, the corresponding semantic model can increase in detail ("saturation level") more so than in size. Additionally, the model can be adapted to accomodate information from multiple documents by inserting and connecting new concepts while tracking their source, thus facilitating multi-document summarization.

Enhancing Expressibility Although they are represented in Fig. 3 as strings, the original expressions used to realize the PSRs recognized by Sparser are stored in the semantic model as parameterized synchronous TAG derivation trees. These trees are used as the "raw material" for realizing the corresponding referents and relationships in text during the generation phase [24]. The set of observed expressions is augmented by a large set of built-in constructions used to realize common semantic relationships such as "is-a" and "has-a," as well as constructions for the types of messages and propositions the SIGHT system

extracts from the graphics. This enables the generation of novel sentences to build an abstractive summary of the extracted information, albeit with some reused and "canned" expressions. Nearly 80% of human-authored summaries are produced using a cut-and-paste method of re-combining original sentences in new ways [18]. Thus, we view our approach as a roughly analogous process at the surface level (except we actually encode the underlying semantic representation), "cutting" semantically-relevant phrases and "pasting" them together with generalized constructions to generate abstract summaries.

3.4 Rating Content

Once the document analysis phase is finished and the semantic model is complete, the model is then analyzed to discover which pieces of information conveyed in the document are most salient. Intuitively, the entries in the model that contain the most important information, and which are highly connected to other important entries, are the ones that ought to be included in the summary. Several factors[5] are used to determine the importance of information extracted from the document and stored in the semantic model:

1. Completeness of attributes: the percentage of filled-in slots for the PSR (i.e., "saturation level"), and the importance of the entries filling these slots (a recursive value)
2. Number of connections/relationships with other PSRs, and the importance of those entries (a recursive value)
3. Number of expressions realizing the referent in the document text (similar to frequency)
4. Salience based on document structure, rhetorical structure, and importance as assessed by the graph summarization algorithm

3.5 Content Selection

Scoring the model based on these factors produces a set of weights for each entry. These weights are passed along to the graph-based content selection framework developed for the SIGHT system [8], which iteratively selects concepts to be conveyed in the summary according to apriori importance, related and redundant information, and discourse history. Using this approach, we are able to include concept completeness, prevalence, and discourse structure captured by the model weighting, as well as incorporate relationship-based centrality assessment.

3.6 Surface Generation

Once the most salient entries in the semantic model have been selected for inclusion in the summary, the surface generation process begins. The previous

[5] Factors 1, 2, and 3 are similar to the dominant slot fillers, connectivity patterns, and frequency criteria proposed by Reimer & Hahn [29].

version of SIGHT [7], which generated descriptions of bar charts only, relied on FUF/SURGE [13] to realize the summaries of graphs in natural language. A large set of templates were used to combine and realize various predicates describing bar charts. However, in order to produce the wider range of constructions necessary to accomodate the article text, and to take advantage of the variety of expressions observed by Sparser and accumulated in the model, the implementation currently in development uses a modern version of Mumble-86 [25] to handle surface realization. For the concepts in the model chosen for inclusion in the summary, we consult the collection of expressions described in Sect. 3.3 and choose from amongst the available options those having the best "fit" (i.e., compatible via substitution or adjunction of TAG trees) enabling these units to be assembled into sentences that are syntactically and semantically valid.

4 Implementation Status

This project is a work in progress and has thus far focused on building the semantic model from text and information graphics. The semantic grammar for Sparser that we have extended is presently capable of producing a nearly-complete parse for several texts in the business news domain (such as the Medtronic article). The SIGHT system is capable of full processing of many simple bar charts (see [10] for limitations), and can identify the intended message in line graphs and grouped bar charts. We are currently working on rating the importance of informational propositions extracted from line graphs, and decomposing these propositions for incorporation into the semantic model. The content rating system remains to be fully implemented and fused with the existing graph-based content selection framework. Finally, a prototype has been developed to use the expressions observed by Sparser for the realization of novel sentences [24], but this component still needs to be integrated with the content rating and selection module. Based on the model built from the Medtronic article, if the resources to be selected by the not-yet-operational content planner are instead chosen by hand, the surface realization component produces the following one-sentence summary:

Wuensch expects a 12-month target of 62 for medical device giant Medtronic.

Company1 ("Medtronic") and Person1 ("Joanne Wuensch," a stock analyst) are the two most prominent concepts in the model (Fig. 2). However, there are no direct links between these concepts, meaning none of the collected phrasings can express them both at the same time. Instead, by using the phrasing provided by a third concept, TargetStockPrice1, we are able to combine all three concepts (via substitution and adjunction operations on the underlying TAG trees), together with a "built-in" realization inherited by the TargetStockPrice category (a subtype of Expectation – not shown in the figure), into the final surface form.

5 Evaluation

Final system evaluation will not be possible until the implementation (in progress) is capable of automatically producing surface output. Summaries generated by

our system will be compared to those of human authors and others created by extractive methods. We will use preference-strength judgment experiments [1] in order to test multiple dimensions of preference (e.g., clarity, completeness). We will also evaluate summaries generated both with and without considering the graphical content, in order to assess the benefits of integrating the contributions of the non-text modalities in the representation of the multimodal document.

6 Future Work

Sparser and KRISP currently require substantial manual effort to build the linguistic and knowledge resources necessary to adapt the system to new domains. Individual grammar rules and ontology definitions must be hand-written by an expert and checked against a corpus of domain texts. Presently, Sparser has decent coverage in the business domain and a few others, but the difficulty of increasing the coverage for broader applications affects scalability. For the implementation currently in development, we are manually extending an existing Sparser grammar on an as-needed basis. While it is relatively trivial to adapt to small changes in an existing domain, adapting to radically-different domains requires a significant amount of resources to be built from the ground-up. In order to automatically adapt the system to new and diverse domains, large-scale learning of additional grammar rules and ontology definitions will be necessary. Promising developments in learning syntactic patterns and ontological relations, as well as machine reading, inspire us to investigate the possibility that these resources may be induced automatically. For example, the Never-Ending Language Learning (NELL) project [5] extracts information from the web in order to extend a structured knowledge base. Similar techniques might be able to build the resources used by our system via automatic domain modeling, with the free-text patterns learned by NELL forming the basis of new Sparser grammar rules.

7 Conclusion

Our approach to automatic summarization of multimodal documents relies on a semantic understanding of text and graphics to construct a unified conceptual model that serves as the basis of generating an abstractive summary. By integrating the knowledge obtained from the graphic with the knowledge obtained from the text at the semantic level, we are able to produce an abstract that treats the entire multimodal document as a single, cohesive message, rather than an assortment of disconnected utterances. This method will generate summaries that are more human-like in nature, while not suffering from coherence and other readability issues related to traditional extractive techniques.

References

1. Belz, A., Kow, E.: Comparing rating scales and preference judgements in language evaluation. In: Proceedings of the 6th International Natural Language Generation Conference. pp. 7–16. INLG 2010, ACL, Trim, Ireland (July 2010)

2. Bhatia, S., Lahiri, S., Mitra, P.: Generating synopses for document-element search. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management. pp. 2003–2006. CIKM '09, ACM, Hong Kong (November 2009)
3. Burns, R., Carberry, S., Elzer, S.: Visual and spatial factors in a bayesian reasoning framework for the recognition of intended messages in grouped bar charts. In: Proceedings of the AAAI Workshop on Visual Representations and Reasoning. pp. 6–13. AAAI, Atlanta (July 2010)
4. Carberry, S., Elzer, S., Demir, S.: Information graphics: an untapped resource for digital libraries. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 581–588. ACM, Seattle (August 2006)
5. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E.R.H., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proc. of the 24th Conference on Artificial Intelligence. pp. 1306–1313. AAAI, Atlanta (July 2010)
6. Chester, D., Elzer, S.: Getting computers to see information graphics so users do not have to. In: Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems, Lecture Notes in Artificial Intelligence 3488. pp. 660–668. ISMIS 2005, Springer-Verlag, Saratoga Springs, NY (June 2005)
7. Demir, S., Carberry, S., McCoy, K.F.: Generating textual summaries of bar charts. In: Proceedings of the 5th International Natural Language Generation Conference. pp. 7–15. INLG 2008, ACL, Salt Fork, Ohio (2008)
8. Demir, S., Carberry, S., McCoy, K.F.: A discourse-aware graph-based content-selection framework. In: Proceedings of the 6th International Natural Language Generation Conference. pp. 17–26. INLG 2010, ACL, Trim, Ireland (July 2010)
9. Demir, S., Oliver, D., Schwartz, E., Elzer, S., Carberry, S., McCoy, K.F.: Interactive SIGHT into information graphics. In: Proc. of the 2010 Int'l Cross Disciplinary Conference on Web Accessibility. pp. 16:1–16:10. ACM, Raleigh, NC (April 2010)
10. Demir, S., Oliver, D., Schwartz, E., Elzer, S., Carberry, S., McCoy, K.F., Chester, D.: Interactive SIGHT: textual access to simple bar charts. The New Review of Hypermedia and Multimedia 16(3), 245–279 (2010)
11. Demner-Fushman, D., Antani, S., Simpson, M., Thoma, G.R.: Annotation and retrieval of clinically relevant images. International Journal of Medical Informatics 78(12), 59–67 (2009)
12. Dumontier, M., Ferres, L., Villanueva-Rosales, N.: Modeling and querying graphical representations of statistical data. Web Semantics: Science, Services and Agents on the World Wide Web 8(2-3), 241 – 254 (2010)
13. Elhadad, M., Robin, J.: An overview of SURGE: a re-usable comprehensive syntactic realization component. In: Proceedings of the 8th International Natural Language Generation Workshop (Posters & Demos). ACL, Sussex, UK (June 1996)
14. Elzer, S., Carberry, S., Chester, D., Demir, S., Green, N., Zukerman, I., Trnka, K.: Exploring and exploiting the limited utility of captions in recognizing intention in information graphics. In: Proceedings of the 43rd Annual Meeting of the Assn. for Computational Linguistics. pp. 223–230. ACL, Ann Arbor (June 2005)
15. Elzer, S., Carberry, S., Zukerman, I.: The automated understanding of simple bar charts. Artificial Intelligence 175, 526–555 (February 2011)
16. Elzer, S., Schwartz, E., Carberry, S., Chester, D., Demir, S., Wu, P.: Bar charts in popular media: Conveying their message to visually impaired users via speech. In: Ras, Z., Tsay, L.S. (eds.) Advances in Intelligent Information Systems, Studies in Computational Intelligence, vol. 265, pp. 275–298. Springer (2010)

17. Fiszman, M., Rindflesch, T.C., Kilicoglu, H.: Abstraction summarization for managing the biomedical research literature. In: Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics. pp. 76–83. ACL, Boston (May 2004)
18. Jing, H., McKeown, K.R.: The decomposition of human-written summary sentences. In: Proc. of the 22nd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. pp. 129–136. ACM, Berkeley (August 1999)
19. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 68–73. ACM, Seattle (July 1995)
20. Lin, C.Y.: Training a selection function for extraction. In: Proceedings of the 8th International Conference on Information and Knowledge Management. pp. 55–62. CIKM '99, ACM, Kansas City (November 1999)
21. McDonald, D.D.: An efficient chart-based algorithm for partial-parsing of unrestricted texts. In: Proceedings of the 3rd Conference on Applied Natural Language Processing. pp. 193–200. ACL, Trento (March 1992)
22. McDonald, D.D.: Issues in the representation of real texts: the design of KRISP. In: Iwańska, L.M., Shapiro, S.C. (eds.) Natural Language Processing and Knowledge Representation, pp. 77–110. MIT Press, Cambridge, MA (2000)
23. McDonald, D.D.: Partially saturated referents as a source of complexity in semantic interpretation. In: Proceedings of the NAACL-ANLP 2000 Workshop on Syntactic and Semantic Complexity in NLP Systems. pp. 51–58. ACL, Seattle (April 2000)
24. McDonald, D.D., Greenbacker, C.F.: 'If you've heard it, you can say it' - towards an account of expressibility. In: Proceedings of the 6th International Natural Language Generation Conference. pp. 185–190. INLG 2010, ACL, Trim, Ireland (July 2010)
25. Meteer, M., McDonald, D., Anderson, S., Forster, D., Gay, L., Huettner, A., Sibun, P.: Mumble-86: Design and implementation. Technical Report 87-87, Dept. of Computer & Information Science, Univ. of Massachusetts (Sept 1987), 174 pgs.
26. Nenkova, A.: Understanding the process of multi-document summarization: content selection, rewrite and evaluation. Ph.D. thesis, Columbia University (January 2006)
27. Radev, D.R., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. Computational Linguistics 28(4), 399–408 (2002)
28. Rau, L.F., Jacobs, P.S., Zernik, U.: Information extraction and text summarization using linguistic knowledge acquisition. Information Processing & Management 25(4), 419 – 428 (1989)
29. Reimer, U., Hahn, U.: Text condensation as knowledge base abstraction. In: Proceedings of the 4th Conference on Artificial Intelligence Applications. pp. 338–344. CAIA '88, IEEE, San Diego (March 1988)
30. Saggion, H., Lapalme, G.: Generating indicative-informative summaries with SumUM. Computational Linguistics 28, 497–526 (December 2002)
31. Srihari, R.K., Zhang, Z., Rao, A.: Intelligent indexing and semantic retrieval of multimodal documents. Information Retrieval 2(2-3), 245–275 (2000)
32. Wu, P., Carberry, S., Elzer, S., Chester, D.: Recognizing the intended message of line graphs. In: Goel, A., Jamnik, M., Narayanan, N. (eds.) Diagrammatic Representation and Inference, LNCS, vol. 6170, pp. 220–234. Springer (2010)
33. Wu, P., Carberry, S., Elzer, S.: Segmenting line graphs into trends. In: Proceedings of the Twelfth International Conference on Artificial Intelligence. pp. 697–703. ICAI 2010, CSREA Press, Las Vegas (July 2008)
34. Yu, H., Liu, F., Ramesh, B.P.: Automatic figure ranking and user interfacing for intelligent figure search. PLoS ONE 5(10), e12983 (10 2010)

Shallow Semantics for Extractive Summarization using Connexor Machinese Semantics

Darren Kipp
School of Information Technology and Engineering
University of Ottawa, Ottawa, Ontario, Canada
Ottawa, Ontario, Canada, K1N 6N5
darrenkipp@gmail.com

Abstract. Connexor Machinese Semantics is one of the most elaborate tools for providing semantic information about a sentence. It has been hypothesized that semantic analysis of sentences is required in order to make significant improvements in automatic summarization. In this work, we look at what shallow semantic features are available that might help to improve the responsiveness of a summary. While this approach is not likely to perform as well as full semantic analysis, it is considerably easier to achieve and could provide an important stepping stone in the direction of deeper semantic analysis.

Keywords: Shallow Semantics, Connexor Machinese Semantics, Extractive Summarization

1 Introduction

The increasing amount of written information which is now available makes it considerably more difficult to find relevant information in an efficient manner. For this reason, it has become necessary to utilize a variety of language tools for searching for documents. While search engines can retrieve entire documents, which may contain relevant sections, they are generally less effective at finding the specific portions of documents which contain the relevant facts and other information required by the user.

Ideally, the user would have available tools which can retrieve only the information relevant to their topic and condense it into a readable form which is also limited in length so that it can be read quickly. This problem, creating a topic-oriented summary, is a difficult natural language processing task. The problem becomes even more difficult as the set of source documents grows.

1.1 Hypothesis/Goal

The goal of this work is to use semantic information to improve responsiveness in automatic multi-document text summarization while keeping other evaluation metrics at approximately the same levels as they have for a system that does not utilize semantic information.

We hypothesize that using some shallow semantic information can improve responsiveness of extractive summaries. This will be done by producing features using a semantic parser/analyzer. Such features can then be added to any extractive summarization system which uses feature vector architecture. For this work we will use Connexor Machinese Semantics [1], a semantic analysis tool, which provides an extensive amount of information about sentences.

Once features have been chosen, we will perform a frequency analysis on the document set, the model summaries and the peer summaries from the 2005 DUC competition (DUC 2005). This frequency analysis will count the number of occurrences of particular features in an attempt to determine if certain features are more likely to appear within summaries. Using this information a final collection of features will be chosen.

Finally, a series of experiments will be conducted in an attempt to utilize these features to improve the responsiveness in topic-oriented multi-document automatic text summarization. Our summaries will be 250 words in length with content extracted from document sets of 30-50 documents each. This is identical to the summarization task at DUC 2005 and several subsequent conferences.

1.2 Measures used for Tuning and Evaluation

There are three main evaluation methods: Pyramid evaluation, ROUGE and manual evaluation.

Pyramid Evaluation is a form of manual evaluation of summaries [2]. It looks only at the content of summaries and not at such phenomena as grammaticality, or style. It is based on comparing the content of the summary with the content of a set of human-written gold standard model summaries.

Summary content units (SCU) are small phrases or snippets found within sentences which are relevant to the topic or question which the summary is based on. These units are created manually from the facts and ideas that appear in human-written model summaries for the topic. The SCUs can then be manually matched with facts and ideas expressed in automated summaries, producing a score. While the initial completion of SCU evaluation for a topic is completely manual, it has been shown that it is largely possible to reverse-engineer the pyramid evaluation to associate SCUs and their weights with sentences in the source documents. A reverse-engineered summary content unit (SCU) corpus has been created for this task thus allowing extractive summarization systems to be trained and tweaked to a pyramid score without the need for human labour [3].

ROUGE, Recall-Oriented Understudy for Gisting Evaluation [4]. is another alternative for automatic evaluation of summaries. It uses several statistical measures to compare a summary with a manually produced model summary. The ROUGE

system is based on a study showing that its scores correlate well with summaries that score well in human evaluation making it unsuitable to use for tuning a system. Sjöbergh [5] demonstrates summaries that would produce high ROUGE scores that would not be considered good summaries by a human reader. This is a general limitation of any automatic statistical evaluation method. Nonetheless, we can use it to evaluate a system tuned by some other means.

Manual evaluation involves a judge or team of judges reading each topic and summary and applying some form of score to a variety of aspects of the summary. We utilize a similar scheme as past DUC competitions. The five measures of linguistic quality were grammaticality, non-redundancy, referential clarity, focus, and structure and coherence. Responsiveness measured the information the summary presented. Manual evaluation is not suitable for tuning since it is labour intensive.

2 Experimental Methodology

A goal of this work is to explore the use of the Connexor semantic parser/analyzer [1] to improve responsiveness in topic-driven multi-document text summarization. A notable source of datasets for text summarization is the Document Understanding Conference's topic-driven multi-document summarization task. For the 2005 and 2006 edition of the conference, extensive submission and evaluation data was available. In this work, the 2005 data was used for tuning and optimizing the new summaries produced by my system while the 2006 will be used exclusively for evaluation. The reverse engineered SCU data file was used to count the total SCU weight within a summary and tune the system. The summarization system described here was manually tuned.

2.1 Choosing Features and Attributes

With the wide variety of features made available by the semantic analyzer, the next requirement is to determine which features would be useful to automatic summarization and how these features could be used. We first consider what features might help chose which sentences to select for inclusion in a summary. Next, we conduct a frequency analysis using 2005 DUC data. The purpose of this analysis is to determine if a feature is present disproportionately in summaries created manually compared with automatic summaries and the source document set. In particular, we are looking for certain types of attributes appearing more frequently in model summaries than in machine generated summaries. With this information, we define a weight (shown in the last column) for combining the feature with lexical matching.

Verb Form. There is a feature in the Connexor Machinese Semantics for the tense of all words tagged as a verb which classifies them as belonging to one of 25 categories. Each of these categories is explained using four hypercategories: past, future, perfect, and progressive. The hypercategories were used to develop features useful for summarization due to the amount of data available.

In the hand written model summaries, there is a noticeably lower proportion of sentences in the future tense. Most topic statements ask for a report on something that has already occurred or for information on the developments in some situation. All of these cases favor information from the past. Due to the structure of verb forms four binary features where created which could be applied to ranking formulas in a variety of ways.

Grammatical Case. The parser identifies three grammatical cases for nouns and pronouns. They are nominative, accusative and genitive. These terms are briefly described by [6]. A noun is often tagged as nominative when it is the subject of a sentence. The accusative nouns usually appear when a noun is the direct object. The genitive case generally appears when the noun or pronoun is showing possession.

An examination of the frequencies of occurrence with the 2005 DUC data reveals the parser did not mark any words as accusative. Within the nominative and genitive classes, the peer summaries selected words at essentially the same frequencies of occurrence as in the document set. The model summaries did slightly favour nominative forms rather then genitive.

Table 1. Frequency analysis of Grammatical Case

Grammatical Case	Document Set		Peer Summaries		Model Summaries		Weight
	Word Count	Percent	Word Count	Percent	Word Count	Percent	
Nominative	306002	97.27%	118668	95.29%	26546	98.11%	+1
Accusative	0	0%	0	0%	0	0%	0
Genitive	8593	2.73%	3194	2.62%	511	1.89%	-1

Person. For the person attribute, we look only at the main verb of a candidate sentence. This avoids possible conflicts with verbs in sub-clauses. There are some noticeable differences in frequency of occurrence within the 2005 DUC document set.

Table 2. Frequency analysis of Person in Main verbs Across Document Sets

Person	Document Set		Peer Summaries		Model Summaries		Weight
	Count	Percent	Count	Percent	Count	Percent	
First	1432	4.4%	132	1.3%	5	0.2%	+2
Second	290	0.9%	88	0.9%	16	0.5%	-5
Third	30599	94.7%	9861	97.8%	3095	99.3%	+4

In all cases, verbs in the 3rd person form are most prominent. Given that the data source was news articles, this makes intuitive sense. In most news articles, first and second person are only used in direct quotes or opinion articles. These types of sentences could be bad for a summary. Sentences in the first person would be a disaster if it is not clear who the speaker is. Likewise, a sentence written in the second person could cause considerable problems since it is not known who the addressee is.

It is notable that within the model summaries very few first and second person sentences are used. This feature will consequently give a slight boost to sentences where the main verb is in the 3rd person, and slightly penalize those where it is not.

Grammatical Degree. The semantic parser marks grammatical degree for adjectives, adverbs, determiners and pronouns. There are three possible classes identified: absolute, comparative and superlative.

Across the DUC 2005 document set, most words were tagged with this attribute were classified as having the absolute degree. The frequencies of words tagged to each class within the peer and model summaries fall along fairly similar lines as frequencies of words tagged to each class in the document set. The model summaries had a slightly higher proportion of absolute degree words and a somewhat lower proportion of superlative degree words.

This makes some intuitive sense because a superlative specifies some sort of extreme or outlier. This may not be desirable in a summary since, in the limited space of a summary, it is usually advantageous to talk about a regular case rather than extreme cases which could simply be exceptions.

Table 3. Frequency analysis of Grammatical Degree

Grammatical Degree	Document Set		Peer Summaries		Model Summaries		Weight
	Count	Percent	Count	Percent	Count	Percent	
Absolute	72281	95.54%	26303	95.29 %	6191	96.25 %	+2
Comparative	1892	2.50%	609	2.21%	154	2.39%	+1
Subjunctive	1479	1.96%	692	2.51%	87	1.35%	-1

Grammatical Mood. Four grammatical moods are identified by the semantic parser: indicative, subjunctive, conditional, and imperative.

Table 4. Frequency analysis of Grammical Mood

Grammical Mood	Document Set		Peer Summaries		Model Summaries		Weight
	Count	Percent	Count	Percent	Count	Percent	
Conditional	809	1.98%	260	2.16%	38	1.01%	+1
Imperative	339	0.83%	97	0.81%	18	0.48%	-2
Indicative	39446	96.61%	11600	96.55%	3685	98.32%	+3
Subjunctive	238	.58%	57	0.47	7	0.19%	+1

The 2005 peers selected sentences with approximately the same frequency of moods as occurred in the source document set. The model summaries marginally favoured sentences with indicative verbs forms. There was a considerable drop in the frequencies of all other moods. In terms of how these terms might apply to a

summary, it is very hard to imagine many situations where imperatives, which usually provide commands or orders, would be useful in a summary.

Sentence Type. Machinese Semantics uses four classes for sentence type: interrogative, declarative, imperative, and exclamative. Across the set of 2005 documents the parser did not identify any sentences as exclamative or imperative. The sentences in the document set and consequently the summaries were mostly declarative sentences with very few interrogative sentences appearing in the human written summaries. Interrogative sentences are all in the form of questions; therefore it is not surprising that few appear in the summaries. The likely reason for this is that the role of the summaries is to provide information or answer some form of direct or indirect question. In many cases, more questions do not provide this type of information.

Table 5. Frequency analysis of Sentence Types

Sentence Type	Document Set		Peer Summaries		Model Summaries		Weight
	Count	Percent	Count	Percent	Count	Percent	
Declarative	42596	99.24%	12780	99.34%	3793	99.71%	+1
Exclamative	0	0%	0	0%	0	0%	-2
Imperative	0	0%	0	0%	0	0%	-2
Interrogative	328	.76%	85	0.66%	11	0.29%	-2

Sentence Function. There are eight classes for sentence type defined by the parser-analyzer. They are statements, commands/directives, exclamations, reporting clauses, tag questions, tone questions, wh-questions, and option questions.

Table 6. Frequency analysis of Sentence Functions

Sentence Type	Document Set		Peer Summaries		Model Summaries		Weight
	Count	Percent	Count	Percent	Count	Percent	
Tag Question	0	0%	0	0%	0	0%	-2
Tone Question	11	0.03%	2	0.02%	0	0%	-2
Wh Question	284	0.77%	61	0.54%	14	0.38%	-2
Op Question	71	0.19%	16	0.14%	3	0.08%	-2
Statement	36324	98.65%	11088	99.02%	3671	99.32%	+2
Command	132	0.36%	31	0.28%	8	0.22%	-2
Exclamation	0	0%	0	0%	0	0%	-2
Reporting Clause	0	0%	0	0%	0	0%	-2

In the frequencies of occurrence in the 2005 data, the question functions do not appear very often in the model summaries relative to the number of times they appear in the document set as well as peer summaries. This is intuitive since it unlikely that

sentences asking questions will provide much information for a summary. Most identified sentence functions were statement. There was also a marginally larger proportion of these used in both the model and peer summaries compared to the document set. Similarly, commands are also less likely to work well in a summary. Command sentences are more useful in giving orders and are harder to conceive as a means of providing information.

Location Correction. A problem with lexical matching is that words that refer to parts of the same thing do not match directly. It is possible, in these situations, to utilize lexical resource such as Wordnet [7] or Roget's Thesaurus [8] to allow such words to match indirectly. The difficulty with such a process is that, applied broadly across a large set of words, it can essentially create too much matching. A summary is limited in size thus adding all synonyms and hyponyms to the matching will simply return a huge set of sentences that can not appear in the summary. It is necessary to be selective expanding lexical matching.

The Connexor semantic analyzer is able to identify location names within sentences. Locations exhibit a hierarchical property, where places are subsets of larger places. For example, the city of Paris is inside France. The result of this is that a topic asking for information or examples within France will not be able to achieve a lexical match with a sentence containing only the word Paris. The semantic analyzer can however identify both Paris and France as locations. We are then able to utilize the Wordnet hierarchy to match these locations together.

Nationality Correction. The nationality correction is very similar to the location correction. An example of a problem that it solves is one where a topic asks for examples of something from within a country, using the nationality rather than the country name. For example: "*Discuss examples of Canadian hydro-electric projects.*"

In such a case the word *Canada* does not produce a match. As well, subset places such as province names or city names also do not help. Here it is necessary to first convert the nationality to its respective country name and then compare the result with subset place names. A difference between this process and the one used to compare locations is that nationalities in both the topic statements and the content sentences must be converted. The conversion between a nationality and a country name are found within Wordnet using holonym relations (member-of relations). This process, like the one for location comparison has the advantage of producing no side-effects in situations where no nationalities are present or the system has failed to match a nationality.

Main Clause. The Connexor semantic parser produces a parse tree which is heavily tagged with information. At the root of the parse tree for a complete parse is the main clause starting with the main verb. In cases of an incomplete parse, a forest is created rather than a single parse tree. There are two sub-cases for incomplete parses. Some incomplete parses will have a main parse tree containing a main clause; however, there are one or more pieces of the sentence which could not be added to the main tree.

The main clause feature is designed to filter out sentences based on how well formed their parse tree is. The value of this binary feature depends on whether a main

verb was found for a candidate sentence. The general idea is that many sentences which do not parse well may not be good sentences anyway so they can be excluded.

Features from Previous Work. In a previous work [3], and [9], there are a number of useful features: number of characters in the sentence, number of words in the sentence, number of the paragraph in which the sentence appears, sentence sequence number in the paragraph, number of discourse connectives [10] in the sentence, number of words in the sentence indicative of causality, number of proper nouns in the sentence, number of content words in the sentence, number of content bigrams, number of punctuation marks in the sentence, and total number of pronouns in the sentence. We will make use of these features as well.

We also add many of the basic features common in many summarization systems including: the number of words, number of characters, paragraph number and sequence in paragraph to the feature set. Similar features have been used in summarization systems dating back as far as 1969 [11].

3 Creating Summaries

The features created from the semantic analyzer output do not combine well using automated mechanisms. A reason for this is that automatic mechanisms for combining features typically use linear or polynomial equations. Breaking away from automatic methods allows for far broader array of methods including filter certain sentences out based on certain features. The disadvantage to using manual tuning is that it becomes impossible to prove an optimal method for combining features. If an optimal combination of features was found it is doubtfully that such a combination would be optimal on a much larger sample of training data. Manual tuning was none the less used because of the limited amount of training data available.

Table 7. Final List of Features Used for Heuristics

Letter	Feature	Letter	Feature
A	Lemmatized Lexical Match	K	Verb Form - Perfect
B	Location Correction	L	Verb Form - Progressive
C	Nationality Correction	M	Person
D	Sentence Function	N	Grammatical Mood
E	Sentence Type	O	Grammatical Case
F	Sentence Too Long	P	Grammatical Degree
G	Sentence Too Short	Q	Total Characters
H	Has Main Clause	R	Total Words
I	Verb Form – Past Tense	S	Paragraph Number
J	Verb Form – Future Tense	T	Position in Paragraph

The simplest formula is to use is the basic lemmatized lexical match feature excluding words appearing on a common list of stop words. This single feature formula also served as a baseline by which to compare other results to. For the 2005 data this produced a total SCU count across all document sets of 107. For the 2006

data the total SCU count was 117 and the normalized value (mean-modified SCU score) was about 0.1738. This alone outperforms a large number of systems submitted at DUC. We then added features to this formula based on their weight values listed in the tables in the previous section.

In combining features and attributes we had the most success with attributes A-H as well as M and S. Our second best combination made use of attributes A-K as well as M and N. It found that while these features all help individually, they often do not improve summaries in combination with each other. This is likely partly due to the limitations of extractive summarization.

4 Summary Evaluation Results

4.1 Manual Evaluation Results

A manual evaluation was performed on a limited amount of data. It permits an accurate evaluation of how good the system is. It also provides the only opportunity to evaluate the linguistic quality of systems. While no direct attempt was made in this work to improve linguistic quality directly, it is still important to evaluate it to ensure that we have not built a system which performs well on responsiveness measures, but is impractical to use due to major problems with readability. For all manual evaluations, the volunteers re-evaluated the DUC submission 23. This submission came from Peking University and the IBM China Research [12]. This was the top scoring system in the mean-modified SCU score evaluation. The limited quantity of data is that it is impossible to calculate statistical significance in any useful way because the confidence intervals become too wide. Consequently we can only compare the average scores. We define the evaluation metrics similar to those at DUC 2006 and 2006.

Table 8. Manual Evaluation Results

System	Respons-iveness	Grammat-icality	Non-Redun-dancy	Referential Clarity	Focus	Structure and Coherence
DUC 2006 System 23	3.375	3.917	3.729	3.770	3.167	2.875
Baseline Lemmatized Lexical Match	3.292	3.333	3.333	3.521	3.000	2.563
New Features in Best Combination	3.042	3.833	3.354	3.333	2.917	2.500

In most manual evaluation measures, the best combination of the new features performed between the baseline Lemmatized Lexical Match and DUC 2006

submission 23. In the case of responsiveness and referential clarity, the best combination of features scored slightly below the baseline measure. A noticeable observation is that the scores for the simple baseline system were very comparable to, and often exceeded, the scores for many of the submissions at DUC. In terms of the responsiveness measure, which the system was designed to improve, all three systems do appear near the top of the rank order list from DUC 2006.

4.2 Automatic Evaluation Results

Summary Content Units. The system built using the semantic information was not the top-performing system, in terms of mean modified SCU score. The 95% confidence interval for the system with semantic information did however contain the mean modified SCU score for the top-performing system at the 2006 DUC challenge. The lack of a statisically significant difference between the systems was most likely due to the lack of data rather than properties of the systems themselves. Many of the differences between in the systems at DUC were very minimal. Even between the best and worst performing systems differences were not large, although they were statistically significant. See table 9 for results.

Table 9. Mean-Modified SCU Scores for DUC 2006 Data

System	Mean Modified SCU Score	Standard Deviation	95% C.I. Lower	95% C.I. Upper
DUC 2006 System 23 (23)	0.242	0.117	0.212	0.273
New Features – Best Combination (bc)	0.211	0.141	0.137	0.250
Baseline Lemmatized Lexical Match (lm)	0.174	0.122	0.137	0.211
DUC 2006 System 33 (33)	0.182	0.092	0.155	0.210

ROUGE.

Table 10. ROUGE-2 Scores for DUC 2006 Data

System	Mean Modified SCU Score	95% C.I. Lower	95% C.I. Upper
DUC 2006 System 23 (23)	0.093	0.079	0.107
New Features – Best Combination (bc)	0.076	0.064	0.088
Baseline Lemmatized Lexical Match (lm)	0.071	0.060	0.083

In the ROUGE evaluation, the system which used the features generated from the semantic parser performed better than the baseline system which used lexical matching. The systems with and without semantic information were within the

statistical confidence intervals of each other. When comparing summaries generated by DUC participants, the system with semantic features ranked in the middle of the list and scored higher than our baseline lexical match system. One possible reason the system did not perform as strongly on the ROUGE measures was that it was not trained using ROUGE. Some systems at DUC have used ROUGE as a training measure. See table 10 for results.

5 Conclusions

Within previous work on automatic text summarization, a number of different approaches have been attempted. These approaches have produced varying results. However, in all cases, these approaches fell short of the results produced when humans manually undertake the task of text summarization. This tends to imply that there is, in general, considerable room to improve automatic summaries. It may, however, require very advanced methods to realize these improvements. These advanced methods may not be easy or even feasible to develop. It is therefore preferable to look for improvements by adding to existing methods and mechanisms. In this work we added a number of features based on semantic information to standard summarization process.

In general, a summarization system will perform best on the evaluation metric it was tuned to. This includes both manual and automatic tuning. For this work, the system was tuned using summary content units (SCUs). Consequently, the system achieved its best score relative to the performance of other systems when this evaluation metric was used. In order to complete an evaluation which is both honest and complete, a number of additional evaluations were performed. In these evaluations, the performance varied, but in general, the performance was not as good as it was in the SCU evaluation.

The addition of some shallow semantic features to query-based summarization systems did not produce dramatic results, but did produce some incite into the types of sentences appearing in summaries. Given the vast amount of information available from a semantic parse, there is significant potential for this type of information to improve query-based multi-document summarization.

The use of the semantic analyzer did not improve responsiveness beyond the level of other high performing methods. There has, however, been a demonstration that the types of features made available by the semantic analyzer could continue to improve summaries if the methods of producing optimal feature sets and methods of combining features could be determined.

6 Future Work

Connexor Machinese Semantics produces a very large number of features, structures and other information useful to summarization and natural language processing. This work found uses and extraction processes for a limited number of them.

Consequently there exists an extensive possibility for the creation of new features, particularly those that dig deeper into the tree-structure.

To assist research on query-based multi-document it would be useful to have additional topics complete with pyramid SCUs. An extension of this is to increase the portion of sentences within topics that are tagged with SCU data. Presently only a small portion of the sentences have any SCU tagging because the others were not selected by any system submitted to the DUC competition. As a result, potentially many useful sentences for selection have no value attached to them

Acknowledgments. I would like to Stan Szpakowicz and Diana Inkpen for their assistance with the work and Terry Copeck for providing his SCU-marked corpus. I would also like to thank my summary evaluators Terry Copeck, Amanda Droeske, Diana Inkpen, Alistair Kennedy, Martin Kipp, Martin Scaiano and Stan Szpakowicz

References

1. Connexor: Connexor Machinese Semantics, http://www.connexor.eu/technology/machinese/machinesesemantics/, Connexor Oy, Helsinki, Finland. (2003).
2. Harnly, A., Nenkova, A., Passonneau, R., & Rambow, O. : Automation of summary evaluation by the pyramid method. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*. Borovets, Bulgaria. (2005).
3. Copeck, T., and Szpakowicz, S.: Leveraging Pyramids. In Proceedings of the Workshop on Automatic Summarization (DUC 2005) at HLT/EMNLP-2005, Vancouver, B.C., Canada. (2005).
4. Lin, C. Recall-Oriented Understudy for Gisting Evaluation (ROUGE), http://haydn.isi.edu/ROUGE/.(2005).
5. Sjöbergh, J.: Older versions of the ROUGEeval summarization evaluation system were easier to fool. In Information Processing and Management: an International Journal, 43(6):1500-1505, November 2007.
6. Loos, E., Anderson, S., Day, D., Jordan, P., Wingate, D.: Glossary of linguistic terms http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/, SIL International. (2004).
7. Miller G.: WordNet: A Lexical Database for English. In *Communications of the ACM*, 38(11):39-41. (1995).
8. Jarmasz, M.: Roget's thesaurus as a lexical resource for natural language processing. Master's thesis, University of Ottawa. (2003).
9. Copeck, T., Inkpen, D., Kazantseva, A., Kennedy, A., Kipp, D., Nastase, V., and Szpakowicz, S.: Leveraging DUC. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 191-197, New York, USA. (2006).
10. Marcu, D.: The Theory and Practice of Discourse Parsing and Summarization. MIT Press. (2000).
11. Edmundson, H.P.: New Methods in Automatic Summarization. Journal of the Association for Computing Machinery, 16(2):264-285, April 1969.
12. Li, S., Ouyang, Y., Sun, B. and Guo, Z.: Peking University at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 101-106, New York, USA. (2006).

Toward Extractive Summarization of Multimodal Documents

Peng Wu and Sandra Carberry

Computer and Information Science Department
University of Delaware,
Newark, DE, 19716
{pwu,carberry}@cis.udel.edu

Abstract. Summarization research has focused on text, and relatively little attention has been given to the summarization of multimodal documents. If extractive summarization techniques are to be used on multimodal documents containing information graphics (bar charts, line graphs, etc.), then a strategy must be devised both for extracting the high-level content of the information graphics and for identifying where that content is relevant in the article's text. This paper gives an overview of our prior work on constructing a summary of an information graphic and presents our new research on methods for selecting paragraphs in a multimodal document that are most relevant to a constituent information graphic. The results demonstrate that our methods are far superior to possible baseline methods and that our work advances the use of extractive techniques for summarizing multimodal documents.

1 Introduction

Summarization research has focused on text, and little attention has been given to multimodal documents. For the most part, this has been due to the difficulty of identifying the content of non-textual components of a document and how this content relates to the document's text. We are addressing the summarization of multimodal documents that consist of text and information graphics, where an information graphic is defined as a non-pictorial graphic such as a bar chart or a line graph. As shown by [2], the message conveyed by an information graphic in popular media (such as newspapers and magazines, as opposed to scientific articles) is often not repeated in the article's text; furthermore, the graphic's caption often contains little or none of the graphic's primary intended message. Thus, information graphics in multimodal documents cannot be ignore.

In previous research[5], we developed a system for constructing a brief summary of information graphics that appear in popular media. One goal of our current research is to extend this work to the summarization of multimodal documents by inserting the graph's summary into the document's text and then applying traditional extractive summarization techniques to construct a summary of the entire document. Unfortunately, unlike scientific articles, the texts

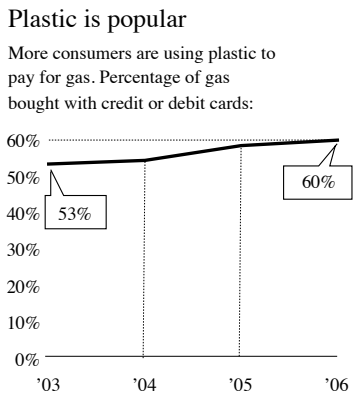

Fig. 1: A line graph from an article about consumer spending

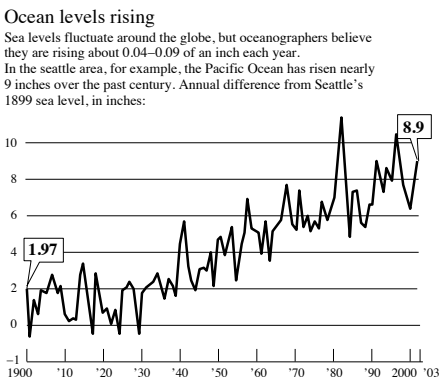

Fig. 2: A line graph from an article about global warming

of multimodal documents from popular media rarely refer explicitly to their information graphics and the graphics often do not appear adjacent to a relevant paragraph (or even on the same page). However, the graph's summary must be inserted at a relevant point in the document. For example, the graph in Figure 1 is included in an article published in *USA Today* with the headline *"Paper or plastic? Answer might save at the pump"*. The most relevant paragraph within the article is the following:

– "More than three-quarters of the gas pumped in the USA is sold at convenience stores. In 2005, 58% of gas was bought using credit and debit cards. Retailers say that number has been climbing in 2006, Lenard says."

But the paragraph closest to the line graph is the following:

– "But on a recent Monday morning, the restaurant owner from Edgemoor, S.C., took out his wallet, went into the gas station convenience store and paid with cash to take advantage of a 4-cent discount for cash customers."

Thus extractive summarization of a multimodal document will lack coherence unless the appropriate placement of content from its information graphics can be identified.

This paper presents our implemented and evaluated methodology for identifying paragraphs in a document that are relevant to an information graphic's content. Section 2 describes our prior work on summarizing information graphics and its relevance to extractive summarization of multimodal documents, along with other important applications of our research. Section 3 then presents our methodology for identifying paragraphs in a document's text that are relevant to an information graphic, Section 4 discusses two examples processed by our system, and Section 5 discusses an evaluation of our methodology. Section 6 discusses related work, and Section 7 describes our future work on the project.

2 Extractive Summarization and Other Applications

Although abstractive summarization is the Holy Grail of summarization research, the state-of-the-art is extractive summarization in which important clauses or sentences are extracted from a document's text. The extracted text is then knitted together into a summary, with the pieces of text generally appearing in the same order as in the original article.

To produce a coherent summary of a multimodal document using extractive summarization techniques, two tasks must be addressed: 1) the construction of a summary of the content of the document's information graphics, and 2) the integration of the graphics' summaries into an overall summary of the document. In previous research, we devised an approach for constructing a summary of an information graphic appearing in popular media. For a line graph, a graph segmentation module first uses a support vector machine to segment the line graph into a sequence of visually distinguishable trends[12]. For example, the line graph in Figure 2 would be converted into two segments, a relatively flat segment from 1900 to 1930 and a rising segment from 1930 to 2003. Then the system extracts communicative signals from the graph, such as whether one bar is colored differently from the other bars, whether a point in a line graph is annotated with its value, or whether a bar label is mentioned in the caption. These communicative signals bring an entity into focus and are used as evidence in a Bayesian network that hypothesizes the graphic's intended message. For example, the intended message of the line graph in Figure 2 is that there is a changing trend in ocean levels — relatively stable between 1900 and 1930 and then rising from 1930 to 2003. The Bayesian network has been implemented for simple bar charts[7] and single line graphs[13]. Next content identification rules (developed from human subject experiments) are used to identify additional propositions that are salient in the graphic and relevant to the graphic's intended message, and these are combined to produce a brief summary of the graphic that is realized in natural language[5].

To produce a summary of a multimodal document containing information graphics, we propose to insert the graph's summary at a relevant point in the article's text and then use extractive summarization techniques to construct a summary of the entire document. But this requires that we identify where the graph's summary should be inserted in the article's text — i.e., which paragraph is most relevant to the information graphic.

In addition to facilitating the application of extractive summarization techniques to multimodal documents that contain information graphics, our work on identifying relevant paragraphs has several other important applications:

1. Our SIGHT system[6] provides blind individuals with access to multimodal documents. SIGHT works within Internet Explorer and uses JAWS screen-reading software. It reads the text of a document to the user; when it encounters an information graphic, it invokes our system to construct a summary of the graphic, which is then relayed to the user via speech. By identifying relevant paragraphs in the document, the effectiveness of the SIGHT system

could be improved by summarizing the graphics at the most appropriate points in the document.

2. We are investigating the indexing and retrieval of information graphics from a digital library. The retrieval methodology will involve a mixture model that takes into account the graphic's intended message, the graphic's textual component such as its caption, and the accompanying textual article. But articles are often long, and much of the article may not be relevant to the information graphic. Thus we hypothesize that our system will perform better if we can identify the paragraphs of the accompanying article that are most relevant to an information graphic and use only these paragraphs in the mixture model that ranks the graphic for retrieval in response to a user query.

3 Methodology for Identifying Relevant Paragraphs

To identify the paragraphs that are most relevant to an information graphic, Section 3.1 proposes a KL divergence based calculation which measures the similarity between the textual component of the line graph and the paragraphs. (The textual component of a line graph consists of three parts: the caption which is the main title for the information graphic, the description which is any additional text that elaborates on the caption, and the "text in graphic" which is any text appearing inside the graphic area.) Section 3.2 then proposes a second method that augments the textual component with words selected from a word list consisting of verbs and adjectives that commonly appear in documents containing information graphics and with the parameters of the intended message of a line graph. The first part of the augmented word list reflects domain-independent graphic content and thus captures words that might appear in a paragraph relevant to *any* information graphic; the parameters of the intended message reflect the line graph's specific content and thus might appear in a paragraph that is specific to this information graphic.

3.1 Method P-KL: KL divergence

Our basic algorithm uses Kullback-Leibler divergence to measure the similarity of two language models, one model for a paragraph in a document and one model for the information graphic's textual component. KL divergence has been widely used in natural language processing and text mining. It measures the difference between two distributions, either continuous or discrete and can be written as

$$D_{KL}(p||q) = \sum_{i \in V} p(i) log \frac{p(i)}{q(i)}$$

where i is the index of a word in vocabulary V, and p and q are two distributions of words. If p and q represent the same word distribution, $D_{KL}(p||q)$ will be 0. For our problem of identifying the relevant paragraphs, p is a smoothed

word distribution built from the line graph's textual component, and q is another smoothed word distribution built from a paragraph in the corresponding document. Smoothing addresses the problem of instances with zero occurrences of a word in the word distribution, which will cause problems in computing the KL divergence. We assign the observed word its true word frequency and assign each unobserved word a low frequency (such as 0.01) and then normalize the word distribution. We rank the paragraphs by their KL divergence score from lowest to highest, since lower KL divergence scores indicate a higher similarity.

3.2 Method P-KLA: KL Divergence with Augmented Textual Component

Our first method only considered the textual component accompanying the line graph. But an information graphic consists of two parts: the textual part and the graphic part. Although the textual part can vary depending on the domain, much of the actual graphic is domain-independent and presents trends, rises or falls, results (higher or lower), or (in the case of bar charts) ranks or comparisons. Thus we decided to explore whether we could automatically extract a set of expansion words that are commonly used in paragraphs that are relevant to information graphics.

To construct this word set, we apply an iterative process in which we automatically identify pseudo relevant paragraphs for each information graphic, extract potential expansion words from the set of pseudo relevant paragraphs identified for all the information graphics, and then repeat the process after augmenting an information graphic's textual component with words from the expansion set. The process is repeated until the expansion word set does not change (convergence) or changes only minimally.

For each information graphic in our training set, we use KL divergence to identify three pseudo-relevant paragraphs in the document. This is similar to the pseudo relevance feedback technology used in information retrieval[15], except that the information retrieval process considers a single query whereas we are using a set of information graphics and associated documents to identify an expansion set that can be applied to all information graphics. If there are N information graphics, we produce a set of 3N relevant paragraphs. The next step is to extract a common word set from the set of pseudo-relevant paragraphs. We assume that the collection of pseudo relevant paragraphs was generated by two models, one producing words relevant to the information graphics and one producing words relevant to the topics of the documents. Let W_g represent the word frequency vector that generates words relevant to the information graphics, W_a represent the word frequency vector that generates words relevant to the domains of the articles, and W_p represent the word frequency vector of the pseudo-relevant paragraphs. We can compute W_p from the pseudo-relevant paragraphs, and we can estimate W_a as the word frequency vector for the entire articles. We want to compute W_g by filtering the components of W_a from W_p. This is similar to the work done by Widdows[11] on orthogonal negation of vector spaces. The problem can be formulated as follows:

1. $W_p = \alpha W_a + \beta W_g$ where $\alpha > 0$ and $\beta > 0$, which means the word frequency vector for the pseudo-relevant paragraphs is a linear combination of the background (topics) word frequency vector and the graphic word vector.
2. $< W_a, W_g >= 0$ which means the background word vector is orthogonal to the graph description word vector. We assume that when the author writes paragraphs that are unrelated to the graphic, he/she will not have the graphic words in mind. Therefore the graphic word vector is independent of the background word vector and these two share minimal information. Since we use a vector space model to represent W_a and W_g, orthogonality is obtained by assuming that these two word vectors have minimum similarity.
3. W_g is assumed to be a unit vector. Whether or not W_g is a unit vector is immaterial for our method, since we are interested only in the relative rank of the word frequencies, not their actual values. However, assuming that W_g is a unit vector gives us three equations in three unknowns (W_g, α, and β) which can be solved for W_g.

With these three assumptions, we obtain

$$\alpha = \frac{< W_p, W_a >}{< W_a, W_a >} \tag{1}$$

$$W_g = \text{normalized}\left(W_p - \frac{< W_p, W_a >}{< W_a, W_a >}\dot{W}_a\right) \tag{2}$$

After we compute W_g, we use WordNet to filter out words whose main sense is neither *verb* nor *adjective*, under the assumption that nouns will be relevant to the domains or topics of the graphs (and are thus *noise*) whereas we want a general set of words (such as *"increasing"*) that are typically used when writing about the data in graphs. To roughly estimate whether a word is predominantly a verb or adjective, we determine whether there are more verbg and adjective senses of the word in Wordnet than there are senses that are nouns.

We then rank the words in the filtered W_g by their frequency and select the k (we chose $k = 25$ in our experiments) most frequent words as our expansion word list. Since the textual components were used to identify pseudo-relevant paragraphs and then pseudo-relevant paragraphs (as opposed to truly relevant paragraphs) were used to construct the word list for expanding the textual components, the accuracy of both the pseudo-relevant paragraphs and the expansion word list are suspect. Thus we apply the two steps (identify pseudo-relevant paragraphs and then extract a word list for expanding the textual components) iteratively until convergence or minimal changes between iterations.

In addition, the parameters of an intended message capture domain-specific content of the graphic's communicative goal. For example, the intended message of the line graph in Figure 2 is ChangingTrend(1900, 1930, 2003) which means that the line graph conveys a changing trend in ocean levels over the period from 1900 to 2003 with the change from relatively stable to rising occurring in 1930. Thus we also added the parameters of the intended message to the augmented word list.

The result is the expansion word list used in method P-KLA. Because the textual component may be even shorter than the expansion word list, we won't add a word from the expansion word list to the textual component unless the compared paragraph also contains this word.

4 Examples

Consider first the graphic in Figure 1. It appeared in an article containing 38 paragraphs. As noted in Section 1, the closest paragraph has little relevance to the graphic. The most relevant paragraph is repeated below:

> *"More than three-quarters of the gas pumped in the USA is sold at convenience stores. In 2005, 58% of gas was bought using credit and debit cards. Retailers say that number has been climbing in 2006, Lenard says."*

Both of our human evaluators selected this paragraph as most relevant to the graphic, and our best performing method, P-KLA, did the same.

Now consider the graphic in Figure 2. This graphic appeared in an article on global warming containing 23 paragraphs. Not only does the paragraph closest to the graphic have little relevance to it, but also no paragraph in the article stands out as overwhelmingly most relevant to the graphic. In fact, the two evaluators selected three and four paragraphs respectively as most relevant, and not only did they differ on their top-ranked paragraph but they also had only one paragraph in common. Although the top-ranked paragraph identified by our best performing method, P-KLA, does not match the paragraph identified as best by either of the human evaluators, the top four paragraphs selected by P-KLA include the four distinct paragraphs identified as relevant by one of the human evaluators. This performance on such a difficult article indicates that our method can handle articles where the most relevant paragraph is not obvious.

5 Evaluation

5.1 The Dataset

We have compiled a dataset of 461 information graphics with full articles from multiple national sources such as *USA Today*, *Business Week*, *News Week*, *New York Times*, and *Wall Street Journal* and some local sources such as *The Wilmington News Journal*. At the time of submission of the final version of this paper, 66 graphs and articles had been analyzed by two human evaluators; thus they were held out as test data and the remainder were used as a training set to build the expansion word list discussed in Section 3.2. For the 66 articles in the test set, the two human evaluators identified paragraphs in each document that were relevant to its constituent information graphic and ranked them in terms of relevance. On average, Evaluator-1 selected 2 paragraphs and Evaluator-2 selected 1.71 paragraphs. For 63.6% of the graphs, the two evaluators agreed on the top ranked paragraph; this shows that in many cases, the most relevant paragraph is not obvious and that several possibilities exist.

5.2 Evaluation Criteria

Both of our methods(P-KL and P-KLA) processed the test set of 66 information graphics with accompanying articles, and each method produced a ranked list of the paragraphs in terms of relevance. We evaluated the results in several ways. For summarization, we want to insert the summary of the graphic at a coherent point in the article's text and then apply extractive summarization on the text. This leads to two evaluation criteria:

1. TOP: the method's success rate in selecting *the most relevant paragraph*, measured as how often the most relevant paragraph identified by the method matches one of the two evaluator's top-ranked paragraph.
2. COVERED: the method's success rate in selecting *a relevant paragraph*, measured as how often the most relevant paragraph identified by the method matches one of the paragraphs identified as relevant by the evaluators.

For our work on retrieving information graphics from a digital library, we want to use several paragraphs of the accompanying article in our mixture model[16] that will rank graphics for retrieval. Thus an appropriate evaluation criteria is normalized discounted cumulative gain (nDCG)[3]. The nDCG is between 0 and 1, and measures how well the rank-order of the paragraphs retrieved by our method agree with the rank-order of the paragraphs identified as relevant by our evaluators. nDCG is defined by the following formulas:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{3}$$

$$\text{where } DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i)} \tag{4}$$

$$\text{and } IDCG_p \text{ is the highest possible } DCG_p \tag{5}$$

We set the cut off position at $p = 3$. The rel_i is the gain of retrieving a paragraph and the $\frac{1}{log_2(i)}$ is the discount according to its position i. The value of rel_i depends on p and the number of relevant paragraphs identified by the human evaluator. If the human evaluator identifies k paragraphs as relevant (where $k \leq p$), then rel_i=k if the i-th ranked paragraph by the system matches the top-ranked paragraph by the human evaluator and is equal to $k - 1$ or $k - 2$ if it matches the paragraph ranked second or third respectively by the human evaluator. Ranking a good paragraph higher gets less discount with the same gain, and ranking a better paragraph at the same position gets higher gain with the same discount.

5.3 Experimental Results

Figures 3 and 4 present the success rate for both of our methods for criteria TOP and COVERED, along with the success rates for two baseline methods: 1) selection of a random paragraph as most relevant, and 2) selection of the

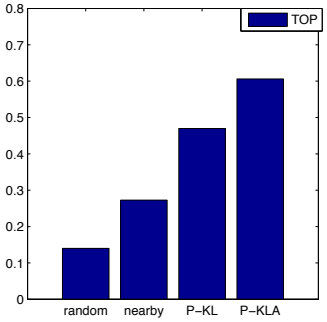

Fig. 3: Success rate in selecting the paragraph identified as most relevant by one of the two human evaluators

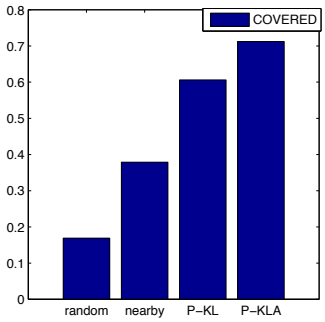

Fig. 4: Success rate in selecting a paragraph identified as relevant by one of the two human evaluators

paragraph that is closest to the information graphic. The results displayed in Figures 3 and 4 show that both of our methods outperform the baseline methods. P-KLA is a further improvement on P-KL. It selects the best paragraph in 60.6% of the test cases, and selects a relevant paragraph in 71.2% of the cases; for both criteria TOP and COVERED, P-KLA doubles or almost doubles the success rate of the baseline methods. The improvement of P-KLA over P-KL indicates that our expansion word list successfully expands the textual component with words pertinent to the graphic itself. A two-sided *student's t-test* shows that the improvements of P-KL over the baseline method and P-KLA over P-KL are both statistically significant at the 0.05 significance level.

Figure 5 presents the results of evaluating both methods in terms of the ranked order of their top three results using nDCG. We measured nDCG using each of the two evaluators as the ideal, and then averaged the results. (When comparing the two human evaluators against one another, their average nDCG is 0.69.) The baseline method in this evaluation is a random selection of three paragraphs from each document. The results in Figure 5 show that all of our methods outperformed the baseline. The best method is P-KLA which more than doubled the baseline method's nDCG. The improvement of P-KLA over P-KL is statistically significant at the 0.05 significance level.

5.4 Using sentence in addition to paragraph to improve the result

Though the paragraph based augmented KL-divergence method gave us satisfactory results, sometimes we consider a paragraph relevant only because there is a relevant sentence in the paragraph, without contribution from other sentences. We hypothesized that taking into consideration both the best sentence in a paragraph and the paragraph itself might further improve the result. We implemented another method named PM-KLA, which computes the final score for a paragraph as a weighted sum of the original score for the paragraph and

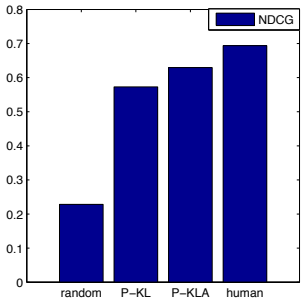

Fig. 5: nDCG scores for the algorithms and human evaluators

Criteria	P-KLA	PM-KLA
TOP	0.606	0.621
COVERED	0.712	0.727
nDCG	0.629	0.655

Table 1: Improved success rate of PM-KLA over P-KLA on three criteria

the score for the best sentence in the paragraph (the sentence with the lowest KL divergence from the augmented textual component).

$$Score_{final_p} = \lambda Score_{\text{best sentence}\in p} + (1-\lambda)Score_p$$

In our experiment, we arbitrarily chose $\lambda = 0.5$. Table 1 shows that the method (PM-KLA) has a higher success rate than P-KLA on both the TOP and COVERED criteria, and a higher nDCG score than P-KLA. However, these improvements are not statistically significant.

6 Related Work

Our work on identifying the paragraph that is most relevant to an information graphic in a multimodal document bears some similarity to the passage retrieval task in text retrieval[10] or question answering[4]. However, we are not doing passage retrieval based on a given query and there is only one document from which we must retrieve a relevant passage. This limits us from using multiple passages retrieved from multiple documents for the same query to improve the result with the relevance feedback technology[9].

Yu et al. [14] used a hierarchical clustering algorithm based on *tf-idf* to associate sentences from an abstract with images in biomedical articles. However, in scientific articles, the image is generally explicitly referred to by a sentence in the article. Thus their method used this referring sentence to identify words relevant to the image, which were likely to be repeated in the sentences of the abstract. In contrast, we are working with articles from popular media which generally

have no such explicit reference to their information graphics; this makes our task more difficult.

A few research efforts have addressed multimodal summarization. Ahmad et al.[1] constructed a system for summarizing financial news and time series data. But instead of summarizing the time series data as text and inserting it into the article, they insert content from the articles into the time series data. Erol et al.[8] combines audio, video and a transcript of the recordings to produce a video summary of a meeting. They use *tf-idf* to identify significant words in the meeting transcript; then they use these words along with features such as intonation in the audio file and high motion in the video recording to identify significant events. These event segments are extracted from the video recording in the order of occurrence and spliced together to produce a video summary of the meeting. This differs from our work in that the different modalities are used only to extract segments from the video recording, whereas we must integrate information extracted from different modalities.

7 Conclusion and Future Work

Summarization is a difficult task, and a multimodal document compounds the problem. Our project's work[5] is the first to construct a summary of the knowledge conveyed by an information graphic, and we are extending this research to the summarization of multimodal documents. This paper addresses a key problem in extractive summarization of multimodal documents containing information graphics — namely, at what point in the document should the content of an information graphic be taken into account in the summary. We have presented methods for identifying the paragraph in the article's text that is most relevant to an information graphic, have analyzed the results produced by each method, and have shown that all of the methods perform far better than any baseline method that might be used. Not only can our best method be used to coherently integrate the content of an information graphic into a summary of a multimodal document, but it can also be used to select passages for use in a mixture model that ranks information graphics for retrieval in a digital library. In future work, we will explore how we might take the graphic's intended message into account when identifying relevant paragraphs and will investigate the quality of extractive summaries of multimodal documents using our approach.

8 Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1016916 and by the National Institute on Disability and Rehabilitation Research under Grant No. H133G080047. We also thank Matthew Stagitis from University of Delaware and Logan Peck from Millersville University for the corpus evaluation work.

References

1. Saif Ahmad, Paulo C F de Oliveira, and Khurshid Ahmad. Summarization of multimodal information. In *the proceeding of LREC 2004*, 2004.
2. Sandra Carberry, Stephanie Elzer, and Seniz Demir. Information graphics: An untapped resource for digital libraries. In *Proceedings of 9th International ACM SigIR Conference on Research and Development on Information Retrieval*, pages 581–588, 2006.
3. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, 2009.
4. Hang Cui, Renxu Sun, Keya Li, Min yen Kan, and Tat seng Chua. Question answering passage retrieval using dependency relations. In *Proceedings of 8th International ACM SigIR Conference on Research and Development on Information Retrieval*, pages 400–407. ACM Press, 2005.
5. Seniz Demir, Sandra Carberry, and Kathleen McCoy. Generating textual summaries of information graphics. In *Proceedings of the International Conference on Natural Language Generation*, pages 7–15, 2008.
6. Seniz Demir, David Oliver, Edward Schwartz, Stephanie Elzer, Sandra Carberry, Kathleen McCoy, and Daniel Chester. Interactive sight: Textual access to simple bar charts. *New Review of Hypermedia and Multimedia*, 16(3):245–279, 2010.
7. Stephanie Elzer, Sandra Carberry, and Ingrid Zukerman. The automated understanding of simple bar charts. *Artificial Intelligence*, 175:526–555, 2011.
8. B. Erol, D.-S. Lee, and J. Hull. Multimodal summarization of meeting recordings. In *Proceedings of International Conference on Multimedia and Expo*, vol. 3, pages 25–28, 2003.
9. Xiaoyan Li and Zhigang Zhu. Enhancing relevance models with adaptive passage retrieval. In *Advances in Information Retrieval*, pages 463–471, 2008.
10. Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Proceedings of the eleventh international conference on Information and knowledge management CIKM '02*, 2002.
11. Dominic Widdows. Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 136–143, 2003.
12. Peng Wu, Sandra Carberry, and Stephanie Elzer. Segmenting line graphs into trends. In *Proceedings of the Twelth International Conference on Artificial Intelligence*, pages 697–703, 2010.
13. Peng Wu, Sandra Carberry, Stephanie Elzer, and Daniel Chester. Recognizing the intended message of line graphs. In *Proceedings of the International Conference on the Theory and Applications of Diagrams*, pages 220–234, 2010.
14. Hong Yu and Minsuk Lee. Accessing bioscience images from abstract sentences. *Bioinformatics*, 22(14):547–556, 2006.
15. Chengxiang Zhai. *Statistical Language Models for Information Retrieval*. Morgan and Claypool Publishers, December 2008.
16. Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 403–410, 2001.

Author Index