Probabilistic Document Modeling for Syntax Removal in Text Summarization

William M. Darling
School of Computer Science
University of Guelph
50 Stone Rd E, Guelph, ON
N1G 2W1 Canada
wdarling@uoguelph.ca

Fei Song
School of Computer Science
University of Guelph
50 Stone Rd E, Guelph, ON
N1G 2W1 Canada
fsong@uoguelph.ca

Abstract

Statistical approaches to automatic text summarization based on term frequency continue to perform on par with more complex summarization methods. To compute useful frequency statistics, however, the semantically important words must be separated from the low-content function words. The standard approach of using an *a priori* stopword list tends to result in both undercoverage, where syntactical words are seen as semantically relevant, and overcoverage, where words related to content are ignored. We present a generative probabilistic modeling approach to building content distributions for use with statistical multi-document summarization where the syntax words are learned directly from the data with a Hidden Markov Model and are thereby deemphasized in the term frequency statistics. This approach is compared to both a stopword-list and POS-tagging approach and our method demonstrates improved coverage on the DUC 2006 and TAC 2010 datasets using the ROUGE metric.

1 Introduction

While the dominant problem in Information Retrieval in the first part of the century was finding relevant information within a datastream that is exponentially growing, the problem has arguably transitioned from finding what we are looking for to sifting through it. We can now be quite confident that search engines like *Google* will return several pages relevant to our queries, but rarely does one have time to go through the enormous amount of data that is supplied. Therefore, automatic text summarization, which aims at providing a shorter representation of the salient parts of a large amount of information, has been steadily growing in both importance and popularity over the last several years. The summarization tracks at the Document Understanding Conference (DUC), and its successor the Text Analysis Conference (TAC)[1], have helped fuel this interest by hosting yearly competitions to promote the advancement of automatic text summarization methods.

The tasks at the DUC and TAC involve taking a set of documents as input and outputting a short summary (either 100 or 250 words, depending on the year) containing what the system deems to be the most important information contained in the original documents. While a system matching human performance will likely require deep language understanding, most existing systems use an extractive, rather than abstractive, approach whereby the most salient sentences are extracted from the original documents and strung together to form an output summary.[2]

In this paper, we present a summarization model based on (Griffiths et al., 2005) that integrates topics and syntax. We show that a simple model that separates syntax and content words and uses the content distribution as a representative model of the important words in a document set can achieve high performance in multi-document summarization, competitive with state-of-the-art summarization systems.

[1] http://www.nist.gov/tac

[2] NLP techniques such as sentence compression are often used, but this is far from abstractive summarization.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 642–647,
Portland, Oregon, June 19-24, 2011.

2 Related Work

2.1 SumBasic

Nenkova et al. (2006) describe *SumBasic*, a simple, yet high-performing summarization system based on term frequency. While the methodology underlying *SumBasic* departs very little from the pioneering summarization work performed at IBM in the 1950's (Luhn, 1958), methods based on simple word statistics continue to outperform more complicated approaches to automatic summarization.[3] Nenkova et al. (2006) empirically showed that a word that appears more frequently in the original text will be more likely to appear in a human generated summary.

The *SumBasic* algorithm uses the empirical unigram probability distribution of the non-stop-words in the input such that for each word w, $p(w) = \frac{n_w}{N}$ where n_w is the number of occurrences of word w and N is the total number of words in the input. Sentences are then scored based on a composition function $CF(\cdot)$ that composes the score for the sentence based on its contained words. The most commonly used composition function adds the probabilities of the words in a sentence together, and then divides by the number of words in that sentence. However, to reduce redundancy, once a sentence has been chosen for summary inclusion, the probability distribution is recalculated such that any word that appears in the chosen sentence has its probability diminished. Sentences are continually marked for inclusion until the summary word-limit is reached. Despite its simplicity, *SumBasic* continues to be one of the top summarization performers in both manual and automatic evaluations (Nenkova et al., 2006).

2.2 Modeling Content and Syntax

Griffiths et al. (2005) describe a composite generative model that combines syntax and semantics. The semantic portion of the model is similar to Latent *Dirichlet* Allocation and models long-range thematic word dependencies with a set of topics, while short-range (sentence-wide) word dependencies are modeled with syntax classes using a Hidden Markov Model. The model has an HMM at its base where one of its syntax classes is replaced with an LDA-like topic model. When the model is in the semantic class state, it chooses a topic from the given document's topic distribution, samples a word from that topic's word distribution, and generates it. Otherwise, the model samples a word from the current syntax class in the HMM and outputs that word.

[3] A system based on *SumBasic* was one of the top performers at the Text Analysis Conference 2010 summarization track.

3 Our Summarization Model

Nenkova et al. (2006) show that using term frequency is a powerful approach to modeling human summarization. Nevertheless, for *SumBasic* to perform well, stop-words must be removed from the composition scoring function. Because these words add nothing to the content of a summary, if they were not removed for the scoring calculation, the sentence scores would no longer provide a good fit with sentences that a human summarizer would find salient. However, by simply removing pre-selected words from a list, we will inevitably miss words that in different contexts would be considered non-*content* words. In contrast, if too many words are removed, the opposite problem appears and we may remove important information that would be useful in determining sentence scores. These problems are referred to as *undercoverage* and *overcoverage*, respectively.

To alleviate this problem, we would like to put less probability mass for our document set probability distribution on non-content words and more on words with strong semantic meaning. One approach that could achieve this would be to build separate stopword lists for specific domains, and there are approaches to automatically build such lists (Lo et al., 2005). However, a list-based approach cannot take context into account and therefore, among other things, will encounter problems with polysemy and synonymy. Another approach would be to use a part-of-speech (POS) tagger on each sentence and ignore all non-noun words because high-content words are almost exclusively nouns. One could also include verbs, adverbs, adjectives, or any combination thereof, and therefore solve some of the context-based problems associated with using a stopword list. Nevertheless, this approach introduces deeper context-related problems of its own (a noun, for example, is not always a content word). A separate ap-

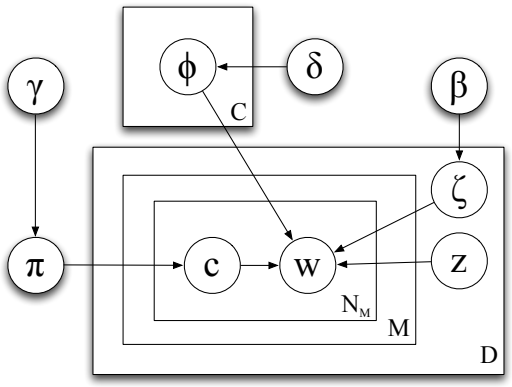

Figure 1: Graphical model depiction of our content and syntax summarization method. There are D document sets, M documents in each set, N_M words in document M, and C syntax classes.

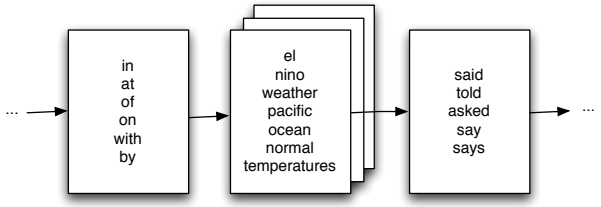

Figure 2: Portion of Content and Syntax HMM. The left and right states show the top words for those syntax classes while the middle state shows the top words for the given document set's content distribution.

proach would be to model the syntax and semantic words used in a document collection in an HMM, as in Griffiths et al. (2005), and use the semantic class as the content-word distribution for summarization.

Our approach to summarization builds on *SumBasic*, and combines it with a similar approach to separating content and syntax distributions as that described in (Griffiths et al., 2005). Like (Haghighi and Vanderwende, 2009), (Daumé and Marcu, 2006), and (Barzilay and Lee, 2004), we model words as being generated from latent distributions. However, instead of background, content, and document-specific distributions, we model all words in a document set as being there for one of only two purposes: a semantic (content) purpose, or a syntactic (functional) purpose. We model the syntax class distributions using an HMM and model the content words using a simple language model. The principal difference between our generative model and the one described in (Griffiths et al., 2005) is that we simplify the model by assuming that each document is generated solely from one topic distribution that is shared throughout each document set. This results in a smoothed language model for each document set's content distribution where the counts from content words (as determined through inference) are used to determine their probability, and the syntax words are essentially discarded.

Therefore, our model describes the process of generating a document as traversing an HMM and emitting either a content word from a single topic's (document set's) content word distribution, or a syntax word from one of C corpus-wide syntax classes where C is a parameter input to the algorithm. More specifically, a document is generated as follows:

1. Choose a topic z corresponding to the given document set ($\mathbf{z} = \{z_1, ..., z_k\}$ where k is the number of document sets to summarize.)

2. For each word w_i in document d

   (a) Draw c_i from $\pi^{(c_{i-1})}$

   (b) If $c_i = 1$, then draw w_i from $\zeta^{(z)}$, otherwise draw w_i from $\phi^{(c_i)}$

Each class c_i and topic z correspond to multinomial distributions over words, and transitions between classes follow the transition distribution $\pi^{(c_{i-1})}$. When $c_i = 1$, a content word is emitted from the topic word distribution $\zeta^{(z)}$ for the given document set z. Otherwise, a syntax word is emitted from the corpus-wide syntax word distribution $\phi^{(c_i)}$. The word distributions and transition vectors are all drawn from Dirichlet priors. A graphical model depiction of this distribution is shown in Figure 1. A portion of an example HMM (from the DUC 2006 dataset) is shown in Figure 2 with the most probable words in the content class in the middle and two syntax classes on either side of it.

3.1 Inference

Because the posterior probability of the content (document set) word distributions and syntax class word distributions cannot be solved analytically, as with many topic modeling approaches, we appeal

to an approximation. Following Griffiths et al. (2005), we use Markov Chain Monte Carlo (see, *e.g.* (Gilks et al., 1999)), or more specifically, "collapsed" Gibbs sampling where the multinomial parameters are integrated out.[4] We ran our sampler for between 500 and 5,000 iterations (though the distributions would typically converge by 1,000 iterations), and chose between 5 and 10 (with negligible changes in results) for the cardinality of the classes set C. We leave optimizing the number of syntax classes, or determining them directly from the data, for future work.

3.2 Summarization

Here we describe how we use the estimated topic and syntax distributions to perform extractive multi-document summarization. We follow the *SumBasic* algorithm, but replace the empirical unigram distribution of the document set with the learned topic distributions for the given documents. This models the effect of not only ignoring stop-words, but also reduces the amount of probability mass in the distribution placed on functional words that serve no semantic purpose and that would likely be less useful in a summary. Because this is a fully probabilistic model, we do not entirely "ignore" stop-words; instead, the model forces the probability mass of these words to the syntax classes.

For a given document set to be summarized, each sentence is assigned a score corresponding to the average probability of the words contained within it: $Score(S) = \frac{1}{|S|}\sum_{w \in S} p(w)$. In *SumBasic*, $p(w_i) = \frac{n_i}{N}$. In our model, *SyntaxSum*, $p(w_i) = p(w_i|\zeta^{(z)})$, where $\zeta^{(z)}$ is a multinomial distribution over the corpus' fixed vocabulary that puts high probabilities on content words that are used often in the given document set and low probabilities on words that are more important in other syntax classes. The middle node in Figure 2 is a true representation of the top words in the $\zeta^{(z)}$ distribution for document set 43 in the DUC 2006 dataset.

4 Experiments and Results

Here we describe our experiments and give quantitative results using the ROUGE automatic text summarization metric for unigram (R-1), bigram (R-2), and skip-4 bigram (R-SU4) recall both with and without (-s) stopwords removed (Lin, 2004). We tested our models on the popular DUC 2006 dataset which aids in model comparison and also on the more recent TAC 2010 dataset. The DUC 2006 dataset consists of 50 sets of 25 news articles each, whereas the TAC 2010 dataset consists of 46 sets of 10 news articles each.[5] For DUC 2006, summaries are a maximum of 250 words; for TAC 2010, they can be at most 100. Our approach is compared to using an *a priori* stopword list, and using a POS-tagger to build distributions of words coming from only a subset of the parts-of-speech.

Method	ROUGE			ROUGE (-s)		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
SB-	37.0	5.5	11.0	23.3	3.8	6.2
SumBasic	38.1	6.7	11.9	29.4	5.3	8.1
N	36.8	7.0	12.2	25.5	4.8	7.3
N,V	36.9	6.5	12.0	24.4	4.4	6.9
N,J	37.4	6.8	12.3	26.5	5.0	7.7
N,V,J	37.4	6.8	12.2	25.5	4.9	7.4
SBH	**38.9**	**7.3**	**12.6**	**30.7**	**5.9**	**8.7**

Table 1: ROUGE Results on the DUC 2006 dataset. Results statistically significantly higher than *SumBasic* (as determined by a pairwise t-test with 99% confidence) are displayed in **bold**.

4.1 SumBasic

To cogently demonstrate the effect of ignoring non-semantic words in term frequency-based summarization, we implemented two initial versions of *SumBasic*. The first, *SB-*, does not ignore stopwords while the second, *SumBasic*, ignores all stopwords from a list included in the Python NLTK library.[6] For *SumBasic* without stop-word removal (*SB-*), we obtain **3.8** R-2 and **6.2** R-SU4 (with the -s flag).[7] With stop-words removed from the sentence scoring calculation (*SumBasic*), our results increase to **5.3** R-2 and **8.1** R-SU4, a significantly large increase. For complete ROUGE results of all of our tested models on DUC 2006, see Table 1.

[4] See `http://lingpipe.files.wordpress.com/2010/07/lda1.pdf` for more information.

[5] We limit our testing to the *initial* TAC 2010 data as opposed to the *update* portion.

[6] Available at `http://www.nltk.org`.

[7] Note that we present our ROUGE scores scaled by 100 to aid in readability.

4.2 POS Tagger

Because the content distributions learned from our model seem to favor almost exclusively nouns (see Figure 2), another approach to building a semantically strong word distribution for determining salient sentences in summarization might be to ignore all words except nouns. This would avoid most stopwords (many of which are modeled as their own part-of-speech) and would serve as a simpler approach to finding important content. Nevertheless, adjectives and verbs also often carry important semantic information. Therefore, we ran a POS tagger over the input sentences and tried selecting sentences based on word distributions that included only nouns; nouns and verbs; nouns and adjectives; and nouns, verbs, and adjectives. In each case, this approach performs either worse than or no better than *SumBasic* using *a priori* stopword removal. The nouns and adjectives distribution did the best, whereas the nouns and verbs were the worst.

4.3 Content and Syntax Model

Finally, we test our model. Using the content distributions found by separating the "content" words from the "syntax" words in our modified topics and syntax model, we replaced the unigram probability distribution $p(\mathbf{w})$ of each document set with the learned content distribution for that document set's topic, $\zeta^{(z)}$, where z is the topic for the given document set. Following this method, which we call *SBH* for "*SumBasic* with HMM", our ROUGE scores increase considerably and we obtain **5.9** R-2 and **8.7** R-SU4 without stop-word removal. This is the highest performing model we tested. Due to space constraints, we omit full TAC 2010 results but R-2 and R-SU4 results without stopwords improved from *SumBasic*'s **7.3** and **8.6** to **8.0** and **9.1**, respectively, both of which were statistically significant increases.

5 Conclusions and Future Work

This paper has described using a domain-independent document modeling approach of avoiding low-content syntax words in an NLP task where high-content semantic words should be the principal focus. Specifically, we have shown that we can increase summarization performance by modeling the document set probability distribution using a hybrid LDA-HMM content and syntax model. We model a document set's creation by separating content and syntax words through observing short-range and long-range word dependencies, and then use that information to build a word distribution more representative of content than either a simple stopword-removed unigram probability distribution, or one made up of words from a particular subset of the parts-of-speech. This is a very flexible approach to finding content words and works well for increasing performance of simple statistics-based text summarization. It could also, however, prove to be useful in any other NLP task where stopwords should be removed. Some future work includes applying this model to areas such as topic tracking and text segmentation, and coherently adjusting it to fit an n-gram modeling approach.

Acknowledgments

William Darling is supported by an NSERC Doctoral Postgraduate Scholarship. The authors would like to acknowledge the financial support provided from Ontario Centres of Excellence (OCE) through the OCE/Precarn Alliance Program. We also thank the anonymous reviewers for their helpful comments.

References

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL 2004: Proceedings of the Main Conference*, pages 113–120. Best paper award.

Hal Daumé, III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312, Morristown, NJ, USA. Association for Computational Linguistics.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. 1999. *Markov Chain Monte Carlo In Practice*. Chapman and Hall/CRC.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Morristown, NJ, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Rachel Tsz-Wai Lo, Ben He, and Iadh Ounis. 2005. Automatically building a stopword list for an information retrieval system. *JDIM*, pages 3–8.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580, New York, NY, USA. ACM.