AUTOMATED NATURAL LANGUAGE HEADLINE GENERATION USING DISCRIMINATIVE MACHINE LEARNING MODELS

by

Akshay Kishore Gattani

B.E.(Honors), Birla Institute of Technology and Science
Pilani (India) 2004

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Spring 2007

APPROVAL

Name: Akshay Kishore Gattani

Degree: Master of Science

Title of project report: Automated Natural Language Headline Generation Using Discriminative Machine Learning Models

Examining Committee: Dr. Greg Mori
Assistant Professor, Computing Science
Simon Fraser University
Chair

Dr. Anoop Sarkar
Assistant Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Martin Ester
Associate Professor, Computing Science
Simon Fraser University
Supervisor

Dr. Fred Popowich
Professor, Computing Science
Simon Fraser University
SFU Examiner

Date Approved: January 18, 2007

Abstract

Headline or short summary generation is an important problem in Text Summarization and has several practical applications. We present a discriminative learning framework and a rich feature set for the headline generation task. Secondly, we present a novel Bleu measure based scheme for evaluation of headline generation models, which does not require human produced references. We achieve this by building a test corpus using the Google news service. We propose two stacked log-linear models for both headline word selection (Content Selection) and for ordering words into a grammatical and coherent headline (Headline Synthesis). For decoding a beam search algorithm is used that combines the two log-linear models to produce a list of k-best human readable headlines from a news story. Systematic training and experimental results on the Google-news test dataset demonstrate the success and effectiveness of our approach.

Keywords: Headline generation, Summarization, Log-linear models, Discriminative Learning, Feature Selection

To my Family

For always supporting my dreams and ambitions

"No matter how good you get, you can always get better and that's the exciting part!"

— *Tiger Woods*

Acknowledgments

It is difficult to express in words my gratitude to Dr. Anoop Sarkar, my Masters supervisor, for allowing me the freedom to pursue my interests and ideas and at the same time ensuring that I never lost sight of my goals. His advice, guidance, encouragement and willingness to help throughout my research made this work possible. I am grateful to Dr. Fred Popowich and my fellow Natural Language lab-mates for creating a wonderful research environment, and especially for the helpful discussions and critical feedback on my work. I would also like to thank my supervisory committee for their constructive criticism on various aspects of this research and report.

I would also like to acknowledge the excellent faculty, especially Dr. Martin Ester, Dr. Ramesh Krishnamurti, Dr. J C Liu, Dr. Greg Mori and the admirable staff of the School of Computing Science for their constant help and support throughout my stay here. A special word of thanks to some wonderful friends I made here: Vinay, Oz, Murray, Fereydoun, Garima and Flavia, for all those Himalayan Peak lunches and for making grad school a fun experience.

Lastly and most importantly, I would like to thank and express my gratitude to my family: my parents for their utmost love and care, my brother for everything he has taught me in life and for motivating me every step of the way, my sister-in-law for being the sweet person she is, my grandma for her blessings, my uncles and aunts for their endearing love and to my cousins and friends back home in India, all of whom I dearly missed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Text summarization has become a driving application of any information or content management system. The explosive growth of world wide web which has mostly unstructured information and online information services has resulted in an information overload problem. Corporations struggle to manage the immense amount of textual information they produce on a day to day basis.

It is not surprising that vast amounts of effort and budget have been devoted both in industry and research towards building automated text summarization systems. Text summarization is the process of distilling the most important pieces of information from given input text or documents and producing abridged versions based on the needs of the task(or tasks) and the user(or users) reading the summary [22]. There are many uses of summarization in everyday scenarios, which are indicative of the types of functions summarization can perform. The different types of summarization tasks can be classified into:

- abstracts (of documents)
- headlines (from news articles around the world)
- table of contents (of a large document)
- outlines (notes for students)
- minutes (of a meeting)
- previews (of movies)

- synopses (soap opera listings)
- reviews (of a book, CD, movie, etc.)
- digests (TV guide)
- biographies (resumes, obituaries)
- abridgments (Shakespeare for children)
- bulletins (weather forecasts/stock market reports)
- sound bites (politicians on a current issue)
- histories (chronologies of salient events)

Within text summarization the focus has almost universally been on extractive techniques i.e. selecting text spans - either complete sentences or paragraphs from the input text. A major pitfall of the extractive summarization techniques is that they cannot generate effective headline styled summaries less than a single sentence or 10 words [2] A special application of text summarization is generating very short summarizes from input text, or headlines from news articles and documents, and is the focus of this work. Headline or headline styled summaries are distinctively different than abstracts of documents. Headlines are terse and convey the singular, most important theme of the input text while abstracts use relatively more words and reflect many important points in the input text.

1.1 Motivation, Applications and Terminology

As mentioned, often the application at hand requires generation of headline styled summaries from text. Such summaries are typically not more than 10-15 words in length. The headline of a text, especially a news article is a compact, grammatical and coherent representation of important pieces of information in the news article. Headlines help readers to quickly identify information that is of interest to them. Although newspaper articles are usually accompanied by headlines, there are numerous other types of news text sources, such as transcripts of radio and television broadcasts and machine translated texts where such summary information is missing. Also in 2003, the Document Understanding Conference (DUC) added the headline generation task to their annual summarization evaluation. The same task was re-introduced in the DUC 2004 conference where short-summary/headline quality was judged based on a set of n-gram overlap metrics called ROUGE [19]. The participation in DUC conferences is again evidence of the growing importance of Headline Generation Systems.

A system that can automatically generate headline styled summaries can be useful in the following potential applications.

- Summarizing emails, web pages for portable wireless devices, WAP enabled mobile phones and PDAs which have limited display and bandwidth.
- Generating a table of contents styled summary for machine generated texts or machine translated documents.
- To present compressed descriptions of search result web pages in search engines
- Headlines extracted from search result web pages can be used to augment a user search query. The resultant query can be used to further re-rank and improve upon the search results. This approach of augmenting a user query with key words extracted from text is being increasingly used in Contextual text search (e.g.: Y!Q [17]) and Information Retrieval.
- From a research standpoint, headline generation offers challenges aplenty in both machine learning and natural language processing.

In the remainder of this thesis, we will use ***headline or headlines*** to refer to titles, headline styled summaries or any short summary of 10-15 words. Also we will refer to documents or any form of textual input collectively as ***news articles*** with the assumption that the size of the document or textual input is reasonable enough to pass as a news article.

1.2 Contributions

The task of headline generation is addressed in two phases:

- **CONTENT SELECTION**: Selecting candidate headline words that reflect the main contents of the article or in other words attributing each word in the news story a probability of its inclusion in the headline. This phase is also referred to as the word selection phase [2]

- **HEADLINE SYNTHESIS**: Combining and ordering word or word phrases to produce a grammatical and coherent headline sentence. This is also referred to as the surface realization phase [2]

The principal contributions of this work are highlighted below.

- We present a discriminative approach for generating headlines from news articles and propose log linear maximum entropy models for both Content Selection and Headline synthesis. The first model captures the notion of what article words should be retained and which ones should be dropped in the headline. The second model captures features of the headline language model and the correlation between words/word phrases that occur both in the headline and the article. The second model enforces grammatical correctness and readability of the headline. We also propose a beam search decoding algorithm that combines the two models to produce a list of k-best human readable headlines. In our model the choice of words in the headline is influenced by both the content selection and headline synthesis models.

- We present a rich feature set for the Content Selection model comprising word and word part-of-speech (POS) n-gram features, word positional features and word frequency based Features that capture both local evidence in the news story as well as the global domain level evidence. Additionally, our Headline Synthesis model combines disparate knowledge sources such as a whole sentence headline Language Model (LM), Headline length (LEN) and an N-Gram Match (MATCH) feature between the headline and news story. These knowledge sources are treated as feature functions and the feature parameters are learnt through state of the art Minimum Error Rate training technique. The flexibility of the framework allows implementers to add additional features and knowledge sources to the model. Our maximum entropy based framework outperforms the baseline statistical headline generation system in [2].

- Previous approaches to headline generation [2] [6] [12] [42] [43] and other extractive techniques impose a restriction on the generated headline by limiting selection of headline words to the words present in the news article. In fact the Naive Bayes approach in [2] further ignores all document words that did not appear in any of the headlines in the training data. Since our framework in feature based with many overlapping features, there is fair competition between all the words in the story for inclusion in the headline. Further by employing a Word Translation model, we also allow words beyond the news article to be present in the headline.

- Our final contribution is in presenting a novel technique for evaluation of a headline generation system that doesn't rely on any form of human assistance or human produced references. Our evaluation method is based on the BLEU metric and makes use of the Google News Service [44] to create benchmark test data sets across multiple domains. The key is that from the Google News service one can readily extract a reference headline set for a certain news article (news event).

1.3 Organization

The remainder of the thesis is organized as follows. In Chapter 2 we discuss previous approaches to headline generation classified into statistical, language based and summarization based schemes. In Chapter 3 we present the theoretical background; mainly the corpus based Machine Learning approach and the Maximum Entropy framework. Chapter 4 is devoted to our headline generation framework, i.e. the Content Selection Model, Headline Synthesis Model and Decoding Algorithm and the feature sets for the two stacked headline generation model. We also discuss parameter estimation techniques for the two models. Chapter 5 presents the experimental setup, results of our technique and comparison against Banko, Mittal and Witbrock's Statistical approach [2]. We present limitations of our work and areas of future work in Chapter 6

Chapter 2

Related Work

Types of headlines can be categorized into *indicative* (headlines that identify the broader topic of the story) and *informative* (headlines that identify the main event or purpose behind the story). Different methods for headline generation typically handle generation of one or the other type of headline. Most previous work on Headline generation can be broadly categorized under Statistical, Rule-based and Summarization based (extractive) approaches. Below we discuss each of these approaches along with the pros and cons of each category. We also present a complete example of each category.

2.1 Summarization based Approaches

One way to connect summarization approaches with headlines is to treat headlines as summaries with a very short length. Given this, we can apply the methods of automatic text summarization to the task of automated headline generation.

In general, extractive approaches towards automatic text summarization can be categorized into three groups: surface-level approaches, entity-level approaches and the combinations of the two. The surface-level approaches find salient sentences for a summary using surface-level features including term frequency [20], the location in text [7], Cue phrases (i.e., phrases indicating the beginning of summary sentences such as "In conclusion", "At the end", etc) [27] [28] and the number of key words or title words in a sentence [7]. Several machine learning algorithms have been proposed for combining these surface level features. Naive Bayes [39], decision trees [8] and semi supervised learning algorithms [15][16] have been examined for combining features. There has been a consensus that, the location of the sentences and presence of cue phrases are more informative than other features. For example researchers have found that simply selecting the lead sentence of the news article as the headline sentence, can be an effective strategy [43].

The entity-level approaches include syntactic analysis, discourse analysis and semantic analysis [33] [23] [37]. These approaches rely heavily on the linguistic analysis of the source text to obtain linguistic structures such as discourse structure, syntactic structure and rhetorical structure to create a summary. There have also been efforts in combining surface level approaches and entity-level approaches together to produce better summaries [9][36].

The advantage of Summarization approaches is that it alleviates the need to treat headline generation as a special problem and one can simply take an existing text summarization system and request it to generate highly compressed summaries as headlines. But the problem with resorting to summarization approaches for headline generation is that, for summarization systems when the compression rate falls below 10%, the quality of generated summaries is poor. Since headlines are typically no more than 10-15 words, the compression ratio is in fact far less than 10% for many news articles. This would mean that text summarization methods will create poor headlines. Another problem with summarization approaches is that most of the techniques we discussed above are extractive in nature which constrains their use in headline generation in other ways. For example: approaches that treat a full sentence as the minimum unit for a summary may result in longer than required headlines. Another problem is that extractive techniques would pick only the phrases and words present in the article for inclusion in the headline. But often we see that headlines do not borrow the exact same words as present in the news article. The example below makes the point clear where the words *attacks* and *fighters* are not present in the article but are used to refer to *the act of striking* and *insurgents* respectively. Given we train our model on sufficiently large corpora we can learn the *attacks* is a good substitution for words like *struck* and that *insurgents* can also be referred to as *fighters.*

Finally, another scenario where extractive summarization approaches are not suitable is cross-lingual headline generation in which news articles are present in one language and headlines need to be generated in a different language. But statistical or corpus based techniques can be used without any specific changes for cross-lingual headline generation just as they are used in the routine scenario. Cross-lingual headline generation can indeed be very useful in cases where say a native language A (English) speaker is looking for language B (French) news articles on a specific topic or event. In such scenarios, the

Headline: NATO attacks Taliban fighters near Kabul

News Article: NATO forces struck suspected Taliban insurgents in
rare violence near Kabul and battles continued Saturday in the area,
NATO said, while an Italian journalist held captive for weeks
returned home, saying he had longed for his family and nation.

Figure 2.1: A sample headline for a news story extract that cannot be generated by Summarization approaches

language A (English) speaker can identify the appropriate language B headlines (French) if corresponding headlines were available in English.

2.1.1 Topiary System

The *Topiary System* was developed by Zajic and Dorr [42][43] at the University of Maryland in association with BBN Technologies. It was the best performing system at DUC 2004 which generated headline styled summaries by combining a set of topic descriptors extracted from the DUC 2004 corpus together with a compressed version of the lead sentence of the news story. The idea behind this approach is that the topic descriptors provide the reader with a general event description while the lead compressed sentence provides a more focussed summary of the news story. The compressed version of the news story is generated using the Hedge Trimmer System which we discuss later in the Linguistic Approaches section, while the topic descriptors are generated using a method called Unsupervised Topic Discovery (UTD). UTD is a statistical method that creates a short list of useful topic labels by identifying commonly occurring words and phrases in the DUC corpus. So for each document in the corpus it identifies an initial set of important topic names for the document using a modified version of the tf.idf metric [35]. Topic models are then created from these topic names using the *OnTopic* software package. The list of topic labels associated with the topic models closest in content to the source document are then added to the beginning of the compressed lead sentence produced in the previous step, resulting in a Topiary-style summary. For example: **BIN LADEN EMBASSY BOMBING: FBI agents this week began questioning relatives**

One of the problems with this approach is that it will only produce meaningful topic models and labels if they are generated from a corpus containing additional on-topic documents on the news story being summarized. As we will see in subsequent chapters, our method identifies candidate headline words *locally* by analyzing the source news article rather than *globally* using the entire corpus, unlike the UTD method. At the same time we also use the tf.idf metric within our CONTENT SELECTION model so that the global nature of the domain/corpus is fed into the local analysis. Topiary can also be categorized as a *hybrid* headline generation model since it employs two very different paradigms into a single framework.

2.2 Statistical Approaches

Statistical or learning approaches assume the availability of a large training corpus (headline-news article pairs) and work in a supervised learning setting. The system or model is trained to learn the correlation between news articles and corresponding headlines and then the learnt model is applied to create headlines for unseen documents. Compared to the Rule-based or summarization based approaches, statistical methods rely on the availability of training data, which can be a disadvantage of these approaches. Also since statistical methods compute the correlation between every news article word and every word in the headline throughout the training data, it is computationally more expensive than summarization or Rule-based approaches. These approaches are ill-suited when there is lack of sufficient training data or when computational resources are limited.

On the other hand, ability to learn from training data is also its strength and adds robustness to these approaches. Unlike Rule-based or some summarization based approaches in which rules for selecting representative sentences and further pruning them to desired length are built into the system, statistical approaches through model training actually learn how to compose a good headline from the training corpus. This ability makes it easier to transport statistical methods to different languages and domains, even making it suitable for cross-lingual headline generation tasks. Also in general, statistical approaches are more robust to noise in the articles making them suitable for producing headlines from machine generated texts. Also statistical approaches can be devised to produce headlines containing words not restricted to the article.

2.2.1 Naïve Bayes Approach

Most current statistical approaches are variations of the Naïve Bayes approach proposed by Banko, Mittal and Witbrock [2]. In the rest of this report we will refer to it as the BMW approach or BMW model. This was the first work to suggest that within a learning framework one could divide the headline generation task into the two phases of content selection and surface realization (headline synthesis). They adopt a Naïve Bayes approach in which the system is trained to learn the correlation between a word in the article D and a word in a headline. They learn the conditional probability of a word appearing in a headline given it appears in the document.

$$P(w \in H | w \in D) = \frac{P(w \in H \wedge w \in D)}{P(w \in D)} \tag{2.1}$$

According to the above expression, one can simply count how many news articles have word w in their headlines and article body and divide it by the number of news articles containing word w in their bodies and use the ratio as the approximation for $P(w \in H|w \in D)$. To enforce the sentence structure and score candidate headlines, i.e. compute the probability of a word sequence S; $P(S)$ they use a bi-gram language model. The overall probability of a candidate summary H consisting of word sequence $(w_1, w_2, ..., w_n)$ is computed as the product of the likelihood of (i) the terms selected for the summary, (ii) the length of the resulting summary, and (iii) the most likely sequencing of the terms in the content set.

$$P(w_1, w_2, ..., w_n|D) = \prod_{i=1}^{n} P(w_i \in H|w_i \in D).P(len(H) = n). \prod_{i=2}^{n} P(w_i|w_1, ..., w_{i-1}) \tag{2.2}$$

In the BMW model $P(w \in H|w \in D)$ is actually an approximation for $P(w \in H|D)$. Thus all evidence to infer whether the word w should be added to a headline or not is based simply on the occurrence of w in the news article. While computing $P(w \in H|D)$ is infeasible because of the infinitely large sample space of the document D, a better approximation than $P(w \in H|w \in D)$ can be arrived upon by considering not just the word occurrence in the article but instead considering the surrounding context of the word along with the word in the news article. We will see in Chapter 4 that an overlapping feature set consisting of word n-grams, word POS, POS n-grams, word tf.idf measure, word position in text and others provides a good 'macro-level' evidence for inference and a better approximation to the content selection model.

Another deficiency of the BMW model is that it constrains the choice of headline words and does not allow words outside of the article to be used as words in the headline. Further, news article words that were never observed in any of the headlines in the training data would have a 0 probability of being included in the headline for a test news article based on this model. In other words, words present in the headlines of the training data are the only words that could be present in the headline of the test data as well, and such a model becomes too restrictive. In chapter 4 we propose techniques for Content Selection that not only give us a better approximation to $P(w \in H|D)$ but also get rid of the above restriction.

Other examples of statistical headline generation techniques are the HMM model proposed by Zajic, et al [42] inverse information retrieval approach [13] and K nearest neighbor approach [12], both by Jin and Hauptmann, and the machine translation model by Kennedy and Hauptmann [14].

2.3 Rule-based Approaches

While similar to Summarization (Extractive) approaches, techniques in this category create a headline for a news story using linguistically motivated heuristics that guide the choice of a potential headline. Hedge Trimmer is an example of this category which uses a parse and trim scheme [6].

The system creates a headline for a news article by removing constituents from the parse tree of the lead sentence of the article until a certain length threshold is reached. Linguistically motivated techniques guide the choice of what constituents should be removed and retained. The principal advantage of these techniques is that they do not require prior training on a large corpus of headline-story pairs since there is no model to be learnt. On the other hand, deciding which single sentence best reflects the contents of the entire news article is a difficult task. Often, news stories have important pieces of information dispersed throughout the article and the approach of trimming the lead or a single important sentence may be unsuccessful in practice. The approach in Hedge Trimmer is very similar to the sentence compression work of Knight and Marcu [16], where a single sentence is shortened using statistical compression. Below we discuss the Hedge Trimmer approach in some detail.

2.3.1 Hedge Trimmer: A Parse and Trim Approach

In the following excerpt from a news story, the words in bold form a fluent and accurate headline for the story. Italicized words are deleted based on information provided in a parse-tree representation of the sentence.

Story Words: **Kurdish guerilla forces** *moving with lightning speed* **poured into Kirkuk** *today immediately* **after Iraqi troops**, *fleeing relentless U.S. airstrikes,* **abandoned** *the hub of Iraqs rich northern* **oil fields**.
Generated Headline: Kurdish guerilla forces poured into Kirkuk after Iraqi troops abandoned oil fields.

Figure 2.2: Example Headline - Hedge Trimmer Approach

For Hedge Trimmer the authors conducted an experiment in which human subjects were asked to create headlines for a corpus of 73 AP stories from the TIPSTER corpus. The only restriction on the produced headline was that the headlines words had to be selected in the order of their appearance in the news story. After examination of distribution of headline words among sentences of the story, they found that 86.8% of headline words were chosen from the first sentence. This distribution is shown in Figure 2.1.

Accordingly, the input to the Hedge Trimmer algorithm is the lead sentence of the news story which is immediately passed through a parser. Later, the following algorithm is used for parse tree trimming.

1. Choose lowest leftmost S with NP,VP
2. Remove low content units
   (a) some determiners
   (b) time expressions
3. Iterative shortening:
   (a) XP Reduction
   (b) Remove preposed adjuncts
   (c) Remove trailing PPs
   (d) Remove trailing SBARs

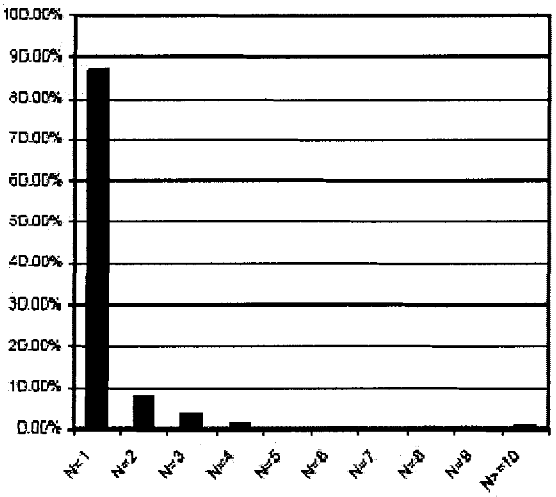


Figure 2.3: Percentage of words from human-generated headlines drawn from Nth sentence of story (taken from [6])

2.4 Classification of Headline Types

Types of headlines (and short summaries) can be categorized into INDICATIVE: headlines which indicate what *topics* are covered by the news story, INFORMATIVE: headlines which convey what particular concept, theme or event is covered in the news story and EYE-CATCHERS: headlines which do not inform about the content of the story but are designed to attract attention and entice people to read the story.

Also an analysis of the three categories discussed earlier reveals that, headline generation approaches and the style of headlines they generate are related. Thus, statistical techniques like the Naïve Bayes (BMW) model is suitable for generating *indicative* headlines, since it scans the entire contents of the news story for selecting headline words. Accordingly, Linguistic techniques like Hedge-Trimmer best generate *informative* headlines since they trim the lead or most important sentence of a headline to an acceptable length. Finally, hybrid approaches like Topiary are a combination of both informative and indicative headlines. Further, depending on the requirements of the headline, one can follow two alternatives: for a high story coverage, statistical methods seem better; for good readability, Rule-based/summarization techniques are better.

2.5 Evaluation of Headlines

Correctly evaluating the machine-generated headlines is an important aspect of automatic headline generation and is a non-trivial task. Relying on human subjects alone to assess the quality of machine-generated headlines i.e. to rank a headline as excellent, good, fair, poor, etc or score it in a range of $1 - 10$ based on different aspects such as grammaticality, relevance and coherance is not full proof. This is because the human judgments for the set of headlines created by one method cannot be used for the evaluation of another set of titles that are generated for the same set of documents but using a different method. Automatic methods for evaluating machine-generated headlines are preferred but are non trivial because various factors such as readability of headlines, quality and consistency of headlines (whether headlines indicate the main content of news story) are hard for a computer program to judge. In particular factors such as quality and consistency can be very subjective and vague to define. In this section we discuss automatic evaluation metrics for headline generation according to factors of readability and consistency. Specifically we discuss 3 metrics: F1, BLEU and ROUGE.

F1 is based on the notion of a system generated headline's Precision and Recall with respect to the reference set of headlines while evaluation metrics BLUE [29] and ROUGE [19] find their roots in Machine Translation Evaluation and have been used in evaluation of headline generation systems. Both BLEU and ROUGE measures make n-gram comparisons of word sequences of machine generated headlines by candidate systems with a set of reference headlines. This set of reference headlines for test data is mostly human generated.

2.5.1 F1 METRIC

$F1$ metric is based on the popular information retrieval notions of *precision* and *recall*. To evaluate headline consistency, i.e. the extent to which the machine-generated headlines are able to capture the contents of documents, word matches between the machine-generated titles and the human assigned titles are measured. As an analogy to information retrieval, the set of machine-selected headline words are treated as the retrieved documents and the set of human selected title words as the marked relevant documents. Therefore, one can easily compute the precision and recall measurement, which have been broadly used in IR. Specifically, for automatic headline generation, the precision of a machine-generated headline with respect to the human-assigned headline is defined as the number of matched words between the machine-generated headline and the human-assigned headline divided by the length of the machine-generated headline. Similarly, the recall of a machine generated headline with respect to the human-assigned headline is defined as the number of matched words between them divided by the length of the human assigned headlines.

However stand alone use of precision or recall is not ideal for measuring word matching. Since, if we only consider the precision metric, a simple strategy to gain the highest precision would be to return the one word sequence *"w"* as the headline where w is the most frequently occurring word in the news story outside of the list of stop words. On the other hand, using recall alone is not good either, because in the extreme case, we can return all the words in the news story and ensure the highest recall. The tradeoff between recall and precision has been well studied in the field of information retrieval, combinations of precision and recall have been found to be effective metrics than the precision and recall alone. The mathematical definition of F1-Metric is as follows.

$$F1 = \frac{2 * precision * pecall}{precision + recall} \tag{2.3}$$

As is evident, F1 metric gives equal emphasis to both precision and recall. When either the precision or the recall is small, the value of F1 will be small. The F1 score is high only when both the precision and recall are large, and will reach the maximum value of 1 only when both the precision and the recall reach their maximum values of 1.

2.5.2 ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a recall-based measure for summarization evaluation. This automatic metric counts the number of n-grams in the reference headlines that occur in the candidate and divides by the number of n-grams in the reference headlines. The size of the n-grams used by ROUGE is configurable. ROUGE-n uses 1-grams through n-grams. Typically there are 6 different ROUGE measures: ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE-LCS and ROUGE-W. The first four metrics are based on the average n-gram match between candidate and reference headlines. ROUGE-LCS calculates the longest common sub-string between the candidate and reference headlines and ROUGE-W is a weighted version of the LCS measure. For all ROUGE metrics, the higher the ROUGE value the better the performance of the summarisation system, since high ROUGE scores indicate greater overlap between the candidate and reference headlines.

Lin and Hovy [18][19] have shown that these metrics correlated well with human judgements of summary quality, and the summarization community has now accepted these metrics as a credible and less time-consuming alternative to manual summary evaluation.

2.5.3 BLEU

BLEU is a system for automatic evaluation of machine translation that uses a modified n-gram precision measure to compare machine translations to reference human translations. This automatic metric counts the number of n-grams in the candidate that occur in any of the reference summaries and divides by the number of n-grams in the candidate. The size of the n-grams used by BLEU is also configurable. BLEU-n uses 1-grams through n-grams. In case of evaluation of headline generation systems, headlines or short summaries are treated as a type of translation from a verbose language to a concise one, and compare automatically generated headlines to a set of human generated headlines. This treatment of headline generation as statistical machine translation is indeed the case for Naïve Bayes (BMW) Model and Zajic, et al's HMM-Hedge model [2][42], both of which treat headline generation as a variant of statistical machine translation.

Specifically, to evaluate the BLEU score for a set of K candidate translations h_K^* against a set of references r_K, we accumulate n-gram precision and closest reference length information for each h_k^* from h_K^* and compute the logarithm of BLEU score as follows:

$$\log BLEU(h_K^*, r_K) = \{\sum_{g=1}^{N} w_g log(p_g) - max(\frac{L_{ref}^*}{L_{sys}} - 1, 0)\} \tag{2.4}$$

where, p_n is the modified n-gram precision which counts the number of n-grams matched between the candidate being evaluated and the reference set and divides this count by the number of n-grams in the candidate. $w_n = 1/N$ where N is the n-gram size we want to consider, typically 3. The second term in the equation is also called the brevity penalty. L_{ref}^* is the effective length of reference headlines and L_{sys} is the effective length of closest reference length matches for the headline candidates.

Chapter 3

Maximum Entropy Framework

Many problems in natural language processing (NLP) can be formulated as classification problems, in which the task is to estimate the correct linguistic "outcome" $a \in A$ given some "context" information $b \in B$. This involves constructing a classifier function $\mathrm{cl} : B \rightarrow A$, which in turn can be implemented with a conditional probability distribution p, such that $p(a|b)$ is the probability of "class" a given some "context" b. Contexts in NLP tasks can vary from fairly simple (single word) to complex (multi-words and associated labels and tags). Large text corpora usually contain some information about the co-occurrence of a's and b's, but never enough to reliably estimate $p(a|b)$ for all possible (a, b) pairs, since the contexts in b are typically sparse. The challenge is then to utilize a method for using the partial evidence about the a's and b's to reliably estimate the probability model p.

Maximum entropy (log linear) probability models offer a clean way to combine diverse pieces of contextual evidence in order to estimate the probability of a certain linguistic outcome occurring with a certain context. We discuss how evidence can be represented in the form of feature functions and explain how to formalize a training problem as a probability model estimation problem under both the Maximum Likelihood framework and the maximum entropy framework. We also discuss that models arrived at by both the methods are essentially the same.

3.1 Feature Functions

We extract evidence presented by training data with the help of feature functions and contextual predicates, a terminology and notation which has become standard when discussing maximum entropy frameworks in NLP [3][32]. Let A be the set of possible outcomes $\{a_1, ..., a_A\}$ and B be the sample space of all possible context information. Then the contextual predicate is a function of the form:

$$cp : B \rightarrow \{true, false\} \tag{3.1}$$

cp evaluates to true or false, corresponding to presence or absence of some information in the context b. In other words, contextual predicates can be thought of as filter functions. Contextual predicates are designed by the experimenter and used in feature functions of the form:

$$f : A \times B \rightarrow \{0, 1\} \tag{3.2}$$

Throughout this work, a feature would be represented as:

$$f_{cp,a'}(a, b) = \begin{cases} 1 & \text{if } a = a' \text{ and } cp(b) = \text{true} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

A feature checks for the co-occurrence of some outcome a' and contextual predicate *cp* evaluating to *true*. The actual set of features for a particular problem is decided by the feature selection strategy and is influenced by the problem domain.

3.2 Corpus Based Approach

In this work we apply a corpus based or machine learning approach in a supervised training setting. Such an approach assumes the existence of a training data set $T = \{(a_1, b_1), \ldots, (a_N, b_N)\}$, which is a large set of pairs of contexts annotated with their outcomes. This training data set is typically extracted using some preliminary form of preprocessing on the textual data. Also it is typical to have binary forms of both the context vectors and outcomes.

3.2.1 Maximum Likelihood Learning

One way to implement a conditional probability distribution to predict outcome a given a context b is to use log-linear or exponential models of the form:

$$p(a|b) = \frac{1}{Z(b)} \prod_{j=1}^{k} \alpha_j^{f_j(a,b)} \tag{3.4}$$

$$Z(b) = \sum_{a'} \prod_{j=1}^{k} \alpha_j^{f_j(a',b)} \tag{3.5}$$

where, k is the number of features and $Z(b)$ is a normalization factor to ensure that the probability of all outcomes sums to 1. Each parameter α_j, where $\alpha_j > 0$, corresponds to a feature f and can be interpreted as a *weight* for that feature. The weights $\alpha_1, ..., \alpha_k$ of the probability distribution p^* that best fits the training data can be obtained by maximum likelihood estimation:

$$\begin{aligned} Q &= \{p|p(a|b) = \frac{1}{Z(b)} \prod_{j=1}^{k} \alpha_j^{f_j(a,b)}\} \\ L(p) &= \sum_{a,b} \tilde{p}(a,b) log(p(a|b) \\ p^* &= \arg\max_{q \in Q} L(q) \end{aligned} \tag{3.6}$$

where, Q is the set of all models of log-linear form, p̃(a b) is the empirical probability of seeing (a, b) in the training set, $L(p)$ is the conditional log-likelihood of the training set normalized by the number of training events, and p^* is the optimal probability distribution according to the maximum likelihood criterion.

3.2.2 Maximum Entropy Learning

The Principle of Maximum Entropy due to Jaynes [11] is based on the premise that when estimating the probability distribution by combining evidences, one should select that distribution that leaves us with the largest remaining uncertainty (i.e., the maximum entropy) consistent with the constraints (evidence) observed in the training data (empirical model). That way we do not introduced any additional assumptions or biases into your model other than what is observed.

Based on a similar corpus based setting, in the conditional maximum entropy framework, the optimal solution p^* is the most uncertain distribution that satisfies the k constraints on feature expectations:

$$\begin{aligned} p^* &= \arg\max_{p\in P} H(p) \\ H(p) &= -\sum_{a,b} \tilde{p}(b)p(a|b)\log p(a|b) \\ P &= \{p|E_p f_j = E_{\tilde{p}} f_j, j = \{1, ..., k\}\} \\ E_{\tilde{p}} f_j &= \sum_{a,b} \tilde{p}(a,b) f_j(a,b) \\ E_p f_j &= \sum_{a,b} \tilde{p}(b)p(a|b) f_j(a,b) \end{aligned} \tag{3.7}$$

$H(p)$ denotes the conditional entropy averaged over the training set, as opposed to the joint entropy, and that the marginal probability of b used here is the observed probability p̃(b), as opposed to a model probability $p(b)$. This is since any model probability p(b) cannot be explicitly normalized over the sample space of possible contexts B, since B is typically very large in practice. $E_p f_j$ is the model p's expectation of f_j, using $\tilde{p}(b)$, the marginal probability. $E_{\tilde{p}} f_j$ denotes the observed expectation of a feature f_j, using $\tilde{p}(a,b)$ the empirical probability of (a,b) in the training data, and P denotes the set of probability models that are consistent with the observed evidence.

There is an important connection between the maximum likelihood and maximum entropy frameworks as both frameworks yield the same answer. Specifically it has been proven that that maximum likelihood parameter estimation for models of form 4.4 is equivalent to maximum entropy parameter estimation over the set of consistent models. That is

$$p^* = \arg\max_{q\in Q} L(q) = \arg\max_{p\in P} H(p) \tag{3.8}$$

3.2.3 Parameter Estimation

One popular method for iteratively estimating the feature parameters in conditional maximum entropy models is Generalized Iterative Scaling (GIS), due to Darroch and Ratcliff [5]. GIS scales the probability distribution $p^{(n)}$ by a factor proportional to the ratio of $E_{\tilde{p}} f_j$ to $E_{p^{(n)}} f_j$. Also in GIS, there is a restriction that the features sum to a constant for any $(a,b) \rightarrow A$, that is

$$\sum_{j=1}^{k} f_j(a,b) = C \tag{3.9}$$

If this condition is not true, it can easily be satisfied with the help of a correction feature by choosing C such that,

$$C = \max_{a \in A, b \in T} \sum_{j=1}^{k} f_j(a,b) \tag{3.10}$$

and correction feature f_l is given by

$$f_l(a,b) = C - \sum_{j=1}^{k} f_j(a,b) \tag{3.11}$$

for any (a,b) pair. Given this setting, the following sequence will converge to p^*:

$$\begin{aligned} \alpha_j^{(0)} &= 1 \\ \alpha_j^{(n+1)} &= \alpha_j^{(n)} [\frac{E_{\tilde{p}} f_j}{E_{p^{(n)}} f_j}] \end{aligned} \tag{3.12}$$

where,

$$\begin{aligned} E_p^{(n)} f_j &= \sum_{a,b} \tilde{p}(b) p^{(n)}(a|b) f_j(a,b) \\ p^{(n)}(a|b) &= \frac{1}{Z(b)} \prod_{j=1}^{k} (\alpha_j^{(n)})^{f_j(a,b)} \end{aligned} \tag{3.13}$$

Given k features, the GIS procedure requires computation of each observed expectation $E_{\tilde{p}} f_j$ once and requires re-computation of the model's expectation $E_p f_j$ on each iteration, for $j = 1, ..., k$. The quantity $E_{\tilde{p}} f_j$ is merely the count of f_j's normalized over the training set:

$$E_{\tilde{p}} f_j = \sum_{a,b} \tilde{p}(a,b,) f_j(a,b) = \frac{1}{N} \sum_{i=1}^{N} f_j(a,b) \tag{3.14}$$

where N is the size of the training sample.

The computation of $E_p f_j$ involves summing over each context b in the training set and each $a \in A$ and dominates the running time of each iteration and the overall procedure.

Chapter 4

Headline Generation Model

In this section we present our headline generation model in detail. In particular, we propose Log-Linear discriminative models for both sentence Content selection and Headline synthesis. Log-Linear or alternatively Maximum Entropy models have been successfully applied in the past to important NLP problems such as parsing [31], Part of Speech (POS) tagging [30], Machine Translation [25], Sentence boundary detection [34], Ambiguity resolution [32] etc. This class of models, also known as log-linear, Gibbs, exponential, and multinomial logit models, provide a general purpose machine learning technique for classification and prediction which has been successfully applied to fields as diverse as computer vision and econometrics. A significant advantage of Maximum Entropy models is that they offer the flexibility and ability to include a rich and complex feature set spanning syntactic, lexical and semantic features. They also offer a clean way to incorporate various information sources and evidences into a single powerful model.

Fig 4.4 depicts the overall framework of our headline generation system. Given a news story, for which k-best headlines are to be generated, the core of the system uses maximum entropy models for both content selection and headline synthesis and a decoding algorithm that explores the space of candidate headline hypotheses to generate the optimal headline word sequences. The decoding algorithm uses the headline synthesis model P_{WS} (equation 4.11) to score candidate headline sequences. The headline synthesis model uses the content selection scores for the words in the sequence as one of the feature functions within the model. The content selection model assigns each word in the news story a probability of its inclusion in the headline. Additionally the headline synthesis model uses four other feature functions over the sequence of words in the form of a language model (P_{LM}), a POS language model (P_{POS_LM}), a headline length feature (P_{LEN}) and a n-gram match model (P_{MATCH}). The decoding algorithm is preceded by a preprocessing phase in which the news story undergoes tokenization, removal of special characters (cleaning) and part of speech tagging. The optimal headline sequences can optionally undergo some form of post processing. For example, verbs in the headline can undergo morphological variation since headline verbs are typically in present tense.

4.1 Content Selection model

Content selection requires the system to learn a model of the relationship between the appearance of some features in a document and the appearance of corresponding features in the headline. The simplest way of modelling this relationship is to estimate the likelihood of some token appearing in a headline given that the token (or possibly a set of tokens) appears in the document to be summarized. Put simply, content selection assigns each document word the probability of being included in the headline. At the same time it should be noted that it is not Content Selection alone, but both Content Selection and Headline Synthesis (Ordering or Realization) that influence whether a word is included in the headline.

Ideally content selection should be modeled as $P(w \in H|D)$ i.e. the probability of word inclusion in the headline given the whole document. But since the sample space of documents or news stories can be infinitely large it is infeasible to learn such a model. Hence various content selection strategies try to approximate computation of $P(w \in H|D)$. One such approximation is the Naïve Bayes content selection model.

$$P(w \in H|D) \approx P(w \in H|w \in D) = \frac{P(w \in H \wedge w \in D)}{P(w \in D)} \tag{4.1}$$

The model can be very easily estimated by counting the number of news articles having word w in their headlines and article body and divide it by the number of news articles containing word w in their bodies. But a better approximation than $P(w \in H|w \in D)$ to content selection can be arrived upon by also considering the context surrounding the word and the word's positional and domain relevant importance (TF*IDF measure) information, since such a model combined evidences from multiple sources.

Formally, for content selection, $p_{CS}(y_w|cx(w))$ denotes the probability of including the word w in the headline, given some contextual information $cx(w)$. $1 - p_{CS}(y_w|cx(w))$ is the probability of not including w in the headline.

Our goal is to build a statistical model $p_{CS}(y'_w|cx(w))$ for content selection that best accounts for the given training data, i.e. a model p which is as close as possible to the empirical distribution $\widetilde{p}$ observed in the training data. A Conditional Maximum Entropy (log linear) model for content selection has the following parametric form.

$$p_{CS}(y_w|cx(w)) = \frac{1}{Z(y_w)} exp[\sum_{i=1}^{k} \lambda_i f_i(y_w, cx(w))] \tag{4.2}$$

where, $f_i(y_w, cx(w))$ are binary valued feature functions that map some form of relationship between the word w and its context $cx(w)$ to either a 0 or 1. λ denotes weights for feature functions. λ_i is the weight for the feature function f_i. The greater the weight, the greater is the feature's contribution to the overall inclusion or non-inclusion probability. k is the number of features and $Z(y_w)$ is the Normalization constant to ensure probability of all outcomes (inclusion and non inclusion of w in headline) sums to 1.

Given the training data, there are numerous ways to choose a model p that accounts for the data. If can be shown that the probability distribution of the form (4.2) is the one that is closest to $\widetilde{p}$ in the sense of minimizing the Kullback–Leibler (KL) divergence between $\widetilde{p}$ and p, when subjected to a set of feature constraints (4.3). The Principle of Maximum Entropy is based on the premise that when estimating the probability distribution, one should select that distribution which leaves you with the largest remaining uncertainty (i.e., the maximum entropy) consistent with the given empirically observed counts.

$$P = \{p|E_p f_i = E_{\widetilde{p}} f_i, i = \{1, ..., k\}\} \tag{4.3}$$

4.1.1 Feature Set for Content Selection

From equation (4.2), the likelihood of a story word being included in the headline depends on the feature vector **f** and corresponding feature weights **w**. Here we present the feature representation $f_{cp,y'_w}(y_w, cx(w))$ over the word w and its surrounding context $cx(w)$. y'_w indicates the inclusion or non-inclusion of w in the headline for values of 1 and 0 respectively. cp is the contextual predicate which maps the pair $< y_w, cx(w) >$ into true or false. Mathematically the feature function can be represented as below.

$$f_{cp,y'_w}(y_w, cx(w)) = \begin{cases} 1 & \text{if } y_w = y'_w \text{ (0/1) and } cp(x) = \text{true} \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

We say that the feature *triggers* if the feature function evaluates to 1, else we say the feature function fails. The word context we consider consists of two words to the left and two words to the right of the word under consideration. The context also includes Part-of-Speech (POS) Tags of all five words, word position information of the current word and the TF*IDF score of the current word. To help understand the subsequent discussion on the feature set consider the following extract from a news story as the example.

"... As/RB much/RB as/IN we/PRP might/MD
[**try/VB to/TO *protect/VB* systems/NNS with/IN**]
new/JJ systems/NNS .... ".

Figure 4.1: News Story Extract showing context of word under consideration

The word under consideration is *protect* and it has a POS tag VB. Also during training, since the stories are accompanied by actual headlines the value of y_w can be determined to be either 0 or 1. To help our discussion let us assume that *protect* is present in the corresponding headline indicating that $y_{protect}$ has a value 1.

4.1.2 Word/Part-of-Speech Features

The first set of features are over adjacent words in the context. These include the current literal token (word), word bi-grams, part-of-speech (POS) bi-grams, the part-of-speech (POS) tri-grams and the POS of each word individually. We consider the word and POS n-gram features both in the forward and backward direction from the word under consideration. These features are meant to indicate likely words to include in the headline as well as provide some level of grammaticality.

The features are explained with examples below. The examples are based on the news story extract in figure 4.1 where w = "*protect*", POS_w = "VB" and $cx(w) = try/VBto/TOprotect/VBsystems/NNSwith/IN$.

Among other information in the context, assume that 1) "protect" is not present in the lead sentence, 2) is present in the top 10% of the news story, 3) has its first occurrence in the top 10% of the news story and 4) has a TF-IDF score in the range of top $10 - 20$.

- **Current Story Word:** This feature triggers if the current word in the feature matches the word in the contextual predicate *cp* and the corresponding outcomes match as well.
  For e.g.: $f_{curr\_word\_is\_protect,1}(1, cx('protect')) = 1$

- **Word Bi-gram Context:** This feature triggers if the word bi-gram in the feature matches the word bi-gram in the contextual predicate *cp* and the corresponding outcomes match.
  For e.g.: $f_{curr\_word\_is\_protect\_curr\_word-1\_is\_to,1}(1, cx('protect')) = 1$
  $f_{curr\_word\_is\_protect\_curr\_word+1\_is\_systems,1}(1, cx('protect')) = 1$

- **POS of Current Story Word:** This feature triggers if the POS tag of current story word equals the POS tag in the contextual predicate *cp* and the corresponding outcomes match. POS tags that are most likely to be included in the headline would get higher weights relative to other POS tags, at the end of model training. For e.g.: POS tags JJ (adjective) or ADV (adverb) are relatively less likely to occur in a headline than POS tags NN or VB.
  For e.g.: $f_{curr\_word\_POS\_is\_VB,1}(1, cx('systems')) = 1$

- **POS Bi-gram of Current Word:** This feature triggers if the POS tag pair of the current story word and previous (next) word equals the POS tag pair in the contextual predicate and the corresponding outcomes match. The intuition is that POS tag pairs that are more common in the headline would get higher weight as a feature at the end of training.
  For e.g.: $f_{curr\_word\_POS\_is\_VB\_curr\_word-1\_POS\_is\_TO,1}(1, cx('protect')) = 1$
  $f_{curr\_word\_POS\_is\_VB\_curr\_word+1\_POS\_is\_NNS,1}(1, cx('protect')) = 1$

- **POS Tri-gram of Current Word:** This feature triggers if the POS tag tuple of current story word and previous (next) word and its previous (next) word equals the POS tag tuple in the contextual predicate and the corresponding outcomes match. Along with POS bi-grams this feature encodes some level of grammaticality in the content selection model.
  For e.g.:

$$f_{curr\_word\_POS\_is\_VB\_curr\_word-1\_POS\_is\_TO\_curr\_word-2\_POS\_is\_VB,1}(1, cx('protect')) = 1$$
$$f_{curr\_word\_POS\_is\_VB\_curr\_word+1\_POS\_is\_NN\_curr\_word+2\_POS\_is\_IN,1}(1, cx('protect')) = 1$$

This set of features is meant to encode lexical tokens and POS contexts that are commonly seen (included) in headlines. However, it should be pointed out that they do so without a larger picture of the function of each word in the sentence. For instance, whether the current word is part of a Noun clause or a Verb clause is not encoded in the context information. Excluding frequently occurring main verbs in the headline is uncommon, since that verb and its arguments typically encode most of the information being conveyed. However words within a relative clause for instance may be dropped from the headline.

4.1.3 Positional Features

The second set of features are based on the positional information of the word in the news story. Experiments and empirical studies have found that a significant proportion of headline words are chosen from the first (lead) sentence of the news story. Similarly in many news stories concluding sentences also convey vital information. Hence the position (in terms of word distance) relative to the beginning of the news story, provides an important cue for its inclusion or non inclusion in the headline. For this we consider the following 3 positional features.

- **Word Position in Lead sentence:** This features triggers if the word under consideration is present in the lead sentence of the story and the corresponding outcomes match.
  For e.g.: $f_{curr\_word\_occurs\_in\_lead,1}(1, cx('protect')) = 0$

- **Word Position:** Additionally, we also divide the news story into the following three intervals based on word count from the beginning.
  - <= 10% - words occurring in the top 10% of the news story
  - >= 90% - words occurring in the last 10% of the news story
  - 10 >< 90% - words occurring in the remainder of the news story not covered by the previous two ranges

  This feature triggers if the current word w and the range matches that of the contextual predicate and the corresponding outcomes match.

For e.g.: $f_{curr\_word\_occurs\_in\_=10\%,1}(1, cx('protect')) = 1$

- **First Word Occurrence Position:** Many headline words, mostly proper nouns frequently repeat throughout the news story. Along with the frequency of occurrence, the position of the first occurrence of the word in the news story provides an important cue. The intervals for this feature are similar to the "Word Position" feature above. This feature triggers if the current word w and the range of first word occurrence matches that of the contextual predicate and the corresponding outcomes match.
  For e.g.: $f_{curr\_word\_\_first\_occurs\_in\_=10\%,1}(1, cx('protect')) = 1$

4.1.4 Word Frequency based Features

This set of features is based on the frequency of occurrence of a word in the document and its frequency of occurrence in the corpus as a whole. Headline words (particularly nouns) tend to occur frequently throughout the news story, at the same time we have to avoid interpreting common words (stop words) such as articles: 'the', 'an', 'a', pronouns: 'this', 'that', etc and prepositions: 'from', 'to', etc as being important based on word frequency. The $tf * idf$ statistical measure as discussed below is precisely the metric to interpret word frequencies in such a manner.

- **Word TF.IDF Range:** This is a novel addition to our feature set and is motivated by information retrieval heuristics. The $TF.IDF$ weight (term frequency inverse document frequency) is a measure often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the $TF.IDF$ weighting scheme are often used by search engines to score and rank a document's relevance given a user query. A high $TF.IDF$ score is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents. We divide the words in the news story into disjoint intervals based on their $TF.IDF$ measure. Based on this division we can say whether the current word has a top 10% $TF.IDF$ measure, $10 - 20\%$ $TF.IDF$ measure and so on. We include a feature function to enable words with high $TF.IDF$ scores to be present in the headline.

The *term frequency* in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term frequency regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document.

$$tf(t_i) = \frac{f_i}{\sum_k f_k} \tag{4.5}$$

Secondly, assume there are N documents in the collection, and that term t_i occurs in n_i of them, then the *inverse document frequency* measure of term t_i is given by

$$idf(t_i) = log\frac{N}{n_i} \tag{4.6}$$

Finally,

$$TF.IDF(t_i) = tf(t_i) * idf(t_i) \tag{4.7}$$

For e.g.: $f_{curr\_word\_tfidf\_in\_10-20,1}(1, cx('protect')) = 1$

- **Stop Word Feature:** As mentioned earlier, due to high frequency of stop words in the news story, they can be deemed as important headline words when actually they are not. So we also include a feature that fires when the considered word is a stop word. Below is a complete list of stop words we consider in this feature.

'a','about','an', 'are','as', 'at','be','by','com',
'for','I','from', 'how','in','is','it',
'of','on','or','that','the','this','to','was',
'when','where','who','will','with', 'the','www'

Table 4.1: List of Stop Words for Content Selection

For e.g.: $f_{curr\_word\_in\_stoplist,1}(1, cx('protect')) = 0$

4.1.5 Potential Considerations

Here we discuss some of the features that although not included in our current implementation of Content Selection Model have proven effective in the area of text summarization.

We have avoided these additional features since they impose extra overhead on both news story preprocessing and on model computation.

- **Cue Phrase feature:** Phrases such as "in summary", "in conclusion", and superlatives such as "the best", "the most important" can be good indicators of important content in a text and have been widely adopted in text summarization systems. One problem with Cue phrases is that they are usually genre dependent. For example, "Abstract" and "in conclusion" are more likely to occur in scientific literature than in newspaper articles. Since newspaper articles often encompass a variety of genres and domains, it is difficult to come up with a general list of Cue phrases for news paper stories. Nevertheless, such a Cue Phrase based feature would trigger if the considered word is present in a news story sentence which has a Cue Phrase in it.

- **Syntax Tree based:** As remarked earlier, our local *Word/Part-of-Speech* features encode lexical and POS based information in the content selection model without the larger function of each word in the sentence.

  [24] is an example of using syntactic evidences as features in sentence compression. It uses deep syntactic analysis of the sentence using the dependency parser and the phrase-structure parser. Such parsers are usually trained out-of-domain and as a result contain noise. Such sentence parse trees help identify important sections within a sentence.

- **Word Net based feature:** A lexical chain is a sequence of related words in the text spanning short (adjacent words or sentences) or long distances (entire text). A chain is independent of the grammatical structure of the text and in effect it is a list of words that captures a portion of the cohesive structure of the text. A lexical chain can enable identification of the concept that the term represents and provide a context for the resolution of an ambiguous term.

  Examples of lexical chains are the following:

  – Rome → *capital* → *city* → *inhabitant*

  – Wikipedia → *resource* → *web*

  Lexical chaining is a method of clustering words in a document that are semantically similar with the aid of a thesaurus like WordNet [45]. Based on word relationships (in order of strength) such as: repetition, synonymy, specialization and generalization, and part/whole relationships one can ascribe a lexical cohesion score to each chained word based on the strength of the chain in which it occurs. Dividing news story words into intervals based on lexical cohesion scores can provide another additional source of information for Content Selection.

Finally, it is important to mention that our current implementation of Content Selection does not stem words to their roots. Using *stemming* and morphological word variations the performance of content selection can be further improved.

4.2 Word translation Model

Consider the news snippet and corresponding headline in Fig 4.2.

```
Headline: Latin states SET TO BACK Panama for U.N. seat

Story: Latin American and Caribbean nations on Thursday
HEADED TOWARD ENDORSING Panama for an open U.N.
Security Council seat after a divisive
battle between U.S.-supported Guatemala and Venezuela.
```

Figure 4.2: News Story Example showing headline word translation

The example highlights a very common feature of how headlines are constructed from news stories by changing the verb word forms. In the above example, *headed toward* is reproduced in the headline as *set to* and *endorsing* as *back*. This reproduction usually happens for words which have a verb POS form and is not typically done for nouns, adjectives and other word forms. Also in the case of headlines, verbs and often action verbs such as *killed*, *destroyed*, etc are very important to essence of the headline. In this case *killed* and *destroyed* could have as well occurred as *shot* and *damaged*. This motivates us to include a word translation probability model for words with verb form so that we have the added flexibility of generating headline words out of the current news story. We define the word translation probability model as:

$$p_{WT}(w_i \in H | w_j \in D) = \frac{p(w_i \in H, w_j \in D)}{p(w_j \in D)} \tag{4.8}$$

where, H is the headline and D is the news story.

The word translation probability p_{WT} in equation 4.8 can be estimated by counting the number of times w_i occurs in headline and w_j occurs in news story in the training data and dividing it with the number of times w_j is observed in the news stories in the training data. This is the maximum likelihood estimation of the word translation probability. Thus for instance if $p_{WT}(killed \in H|shot \in D)$ has a high probability relative to other possible substitutions for "shot" then we can substitute "shot" with "killed" in the headline with high confidence.

Now let W_{vb} be the set of all words that have verb forms and that occurred in any of the headlines in the training data. Combining the Content selection model and Word Translation model, for any $w_i \in W_{vb}$ we could write,

$$p_{CS}(y_{w_i} \in H|w_j \in D) = p_{WT}(w_i \in H|w_j \in D) p_{CS}(y_{wj} \in H|w_j \in D) \tag{4.9}$$

In other words, we just multiply the content selection probability of word w_j with the probabilistic weight with which another word w_i can be substituted for it. As we will see later on this model allows us to expand word choices for the content selection model beyond the default bag of words present in the news story. The above equation for calculating content selection probabilities of substituted words is based on the assumption that content selection of word w_i is dependent only on w_j's word translation model and is independent of contents of the news story or the actual content selection model for the news story.

The following toy example explains the use of word translation probability. Consider the content selection probability of a particular instance of the word "nations" in a news story D to be 0.30, i.e. $P_{CS}(y_{\text{"}nations\text{"}}|D) = 0.30$. Then the following table gives the new content selection probabilities of possible substitutions of the word "nations".

Substitution-w_i for w_j = "*nations*" ($\mathrm{P}_{WT}(w_i\|w_j) > 0$)	$\mathrm{P}_{WT}(w_i\|w_j)$	$\mathrm{P}_{CS}(y_{wi}\|D)$
"states"	0.20	$0.20 * 0.30 = 0.06$
"countries"	0.10	$0.10 * 0.30 = 0.03$
"..."	...	...

Table 4.2: Example showing use of Word Translation Model

4.3 Headline Synthesis Model

In our framework for headline generation, the process of headline generation is divided into two phases, the phase of finding good headline words for a news story and the phase of organizing selected headline words into sequences. While the first conditional maximum entropy model takes care of the first phase: Content Selection, the second conditional maximum entropy model takes care of the second phase: Headline synthesis.

Headline Synthesis or alternatively Surface Realization assigns a score to the sequence of surface word ordering of a particular headline candidate by modeling the probability of headline word sequences in the context of the news story. [2] uses the simplest form of word ordering model in the form of a bi-gram language model to reasonable effect. In [2], the probability of a word sequence is approximated by the product of the probabilities of seeing each term given its immediate left context. A major drawback of using a bi-gram language model for scoring headline candidates is that it fails to consider any context provided by the news story as the scoring is done on a word sequence independent of the contents of the news story.

Our headline scoring function is motivated by the use of Maximum Entropy models in Statistical Machine translation proposed by Och and Ney [25] in which the best performing statistical models combine different models or knowledge sources (translation model, language model, alignment model, etc) using maximum entropy parameter estimation. The popular source-channel approach in Machine Translation is contained as a special case in the Maximum Entropy Model.

In this framework, for example given a French source language sentence f_1^J and a candidate English translation e_1^I, the machine translation probability is given by

$$Pr(e_1^I|f_1^J) = p_{\lambda_1^M}(e_1^I|f_1^J) = \frac{exp[\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)]}{\sum_{e_1'^I} exp[\sum_{m=1}^{M} \lambda_m h_m(e_1'^I, f_1^J)]} \tag{4.10}$$

In this framework, there are a set of M feature functions and for each feature there exists a model parameter λ_m, $m = 1, 2, ..., M$ for each feature.

We extend a similar idea to our headline generation framework. Our headline scoring function takes into consideration 5 main aspects (or information sources) of the generated sequence of words. These information sources are contained within a conditional maximum entropy model over the sequence of words, that evaluates the candidate headline sentences and assigns each candidate a score. Given a sequence of words $H = w_1, w_2, ..., w_n$ as a candidate headline for a news story D, the Whole Sentence headline synthesis model is given by

$$p_{WS}^{\alpha_1^M}(H = w_1, w_2, ..., w_n|D) = \frac{exp[\sum_{m=1}^{M} \alpha_m h_m(H = w_1, w_2, ..., w_n)]}{\sum_{H'} exp[\sum_{m=1}^{M} \alpha_m h_m(H' = w_1', w_2', ..., w_n')]} \tag{4.11}$$

In this model, there are $M = 5$ feature functions and for each feature there exists a model parameter α_m, $m = 1, 2, ..., M$. The 5 feature functions (information sources) in this maximum entropy model are as below.

- **Language Model Feature**: A bi-gram language model trained on a training corpus consisting only of headline sentences is used to assign the sequence of words in the candidate headline a score as below.

$$h_1(H = w_1, w_2, ..., w_n) = h_{LM}(H = w_1, w_2, ..., w_n) = \sum_{i=2}^{n} \log(P(w_i|w_{i-1})) \tag{4.12}$$

- **POS Language Model Feature**: A Part-of-Speech tri-gram language model trained on a Part-of-Speech annotated training corpus consisting only of headline sentences is used to assign the sequence of Part-of-Speech of words in the candidate headline a score as below.

$$\begin{aligned} h_2(H = w_1, w_2, \ldots, w_n) &= h_{POS\_LM}(H = w_1, w_2, \ldots, w_n) \\ &= \sum_{i=3}^{n} \log(P(POS_{w_i}|POS_{w_{i-1}}, POS_{w_{i-2}})) \end{aligned} \tag{4.13}$$

- **Headline Length Feature**: A headline length probability distribution is computed on the training corpus consisting of headline sentences to assign the length of candidate headline sentence a probability. This feature biases the length of generated candidate headlines to be within typical observed lengths of 5 to 15 words and penalizes headlines with too short or too long a length.

$$h_3(H = w_1, w_2, \ldots, w_n) = h_{LEN}(H = w_1, w_2, \ldots, w_n) = \log(P(len(H) = n) \tag{4.14}$$

- **Content Selection Feature**: Content selection scores that were computed for each story word during the content selection phase are also critical information sources. Words with high CS scores would bias the headline synthesis model into choosing sequences with words having high content selection probabilities. The content selection feature score is the sum of log probabilities of content selection probabilities of each word in the sequence.

$$h_4(H = w_1, w_2, ..., w_n) = h_{CS}(H = w_1, w_2, ..., w_n) = \sum_{i=1}^{n} \log(P_{CS}(w_i|D)) \qquad (4.15)$$

- **N gram Match Feature**: In order that the headline synthesis model does not choose words from disparate sections and sentences of the news story and to ensure that word sequences in the headline maintain some continuity with respect to the news story, we have added a word N-gram match feature where the value of N is 3. N-gram match feature is essentially calculating the logarithm of the BLEU score of the headline with respect to the news story instead of reference headline sentences. Also there is no brevity penalty in this case of calculating N gram headline match with respect to the news story.

$$h_5(H = w_1, w_2, ..., w_n) = h_{MATCH}(H = w_1, w_2, ..., w_n) = \log \sum_{n=1}^{N} \gamma_n log(p_n) \qquad (4.16)$$

where, γ_n are positive weights summing to one. In our system we use, N = 3 and uniform weights $\gamma_n = 1/N$. p_n is the N-gram precision using n-grams up to length N, where precision is calculated with respect to the news story.

Language model features 1 and 2 enforce grammaticality over the sequence of words, feature 3 penalizes candidate headlines straying away from typical headline lengths (usually between 5 and 15), feature 4 ensures accuracy of words included in the headline through the content selection model, while feature 5 in conjunction with feature 4 enforces coherence and continuity in the generated candidate headlines, since contiguous words sequences (word phrases) would be encouraged during selection.

4.4 Decoding Algorithm

As mentioned before, inclusion of a particular word in the headline is influenced by both the models: content selection and headline synthesis. The decoding algorithm incrementally builds sequences from left to right in the form of candidate headline hypothesis and also systematically combines the content selection model and the headline synthesis model together. To find the optimal candidates (paths) we use a Beam search algorithm with pruning. Beam search algorithms have also been widely adopted in Statistical Machine Translation (SMT) systems [38].

The search is performed by building partial sequences (hypotheses), which are stored in a priority list (or lists). Word sequences are scored based on the *whole sentence headline synthesis model* in equation 4.10. Content selection probabilities are indirectly fed as scores into this model. The priority list helps retain only promising sequences with very high scores and discards all other hypotheses with low scores to make the search feasible.

Our decoding or headline search algorithm considers as input a news article with the word sequence $w_1, w_2, ...w_L$ and the word POS sequence $p_1, p_2, ...p_L$, where L is the length of news story. T, the number of top scoring headlines returned by the algorithm is user input. Other inputs to the algorithm are the Content Selection model (parameters): λ, the headline synthesis model parameters: α, the word translation model: P_{WT} and the individual model components of the headline synthesis model: $P_{LM}, P_{POS\_LM}, P_{LEN}$. TF-IDF scores and N-gram match scores are computed on the fly, while the IDF component of TF-IDF is computed beforehand on the training corpus.

The headlines are ranked by the *Whole-Sentence Headline Scoring Function* using equation 4.11 at the end of each iteration and pruned down to maximum of C candidates, in other words C is the hypotheses cut-off. One problem when building a beam search decoder is that decoders tend to bias the search towards those sequences that had higher probabilities during the first stages (initial iterations). This is not always the best scenario, and in order to ensure that all early hypotheses receive fair comparison and compete to stay alive after pruning, we maintain two separate cut offs; C_1 and C_2. C_1 is used during the initial iterations (0-5) and C_2 is used during subsequent iterations ($6 - ML$). ML is the maximum length of headline word sequences and is the last iteration.

The values for both C_1 and C_2 were tuned to 20 and 10 respectively in our system. While ML is set to 15 as is typical of headline lengths. Alternative approaches to pruning can also be used, such as keeping all hypotheses after increments which have scores lying within a certain radius of the score of the original hypothesis before the increments are done. Additionally, to account for word substitutions due to word translation model we initialize W_{vb} to be the set of all words that have verb forms and which occur in any of the headlines in the training data. W_{vb} is further pruned down during the content selection phase in presence of the word translation model using Algorithm 2. The complete algorithm is given in Algorithm 1.

4.5 Parameter Estimation

Given the general forms of both Content Selection and subsequent Headline Synthesis models, we still need to estimate parameter vectors λ and α respectively from the training data. The content selection model is comprised of hundreds of thousands of features, hence the parameter vector λ governing the content selection model is trained on large amounts of training data to avoid over-fitting as much as possible. α on the other hand is the vector of model scaling parameters and these weights are tuned on a much smaller model scaling data set. Below we discuss techniques for estimation of parameters for both models.

4.5.1 Contention Selection Parameters: λ

For estimating weights of parameter λ of the content selection model, we use the Generalized iterative scaling method as discussed in Chapter 3. Specifics of the GIS method implementation are discussed in the Experimental Evaluation Section 5.2.

4.5.2 Headline Synthesis (Model Scaling) Parameters: α

Choosing the parameters of this exponential model or equivalently *scaling factors* for each knowledge source (component model) in the headline synthesis model significantly

Algorithm 1 Algorithm to generate top T headlines using Beam Search Decoding

```
Require: News story words $D = \{w_1, ..., w_l, ..., w_L\}$, POS of news story words $P = \{p_1, p_2, ...p_L\}$, number of top scoring headlines to be returned $T$, beam search cutoffs $C_1, C_2 > T$ $(C_1 > C_2)$, Maximum length of headline $ML$, initial word translation set $W_{vb} = \{v_1, \ldots, v_j, ..., v_V\}$, content selection parameter list $\alpha$, headline synthesis parameter list $\lambda$, Models $P_{WT}$, $P_{LM}$, $P_{POS\_LM}$ and $P_{LEN}$.
Ensure: List of Top $T$ Candidate headlines; $h^1, ..., h^T$, where each $h^m = h^m_1, ..., h^m_{ML}$
# Initialize priority queues
$S = S_f = []$
# Initialize word translation substitution set
$D_v = Calculate\_Word\_Translation\_Set( D, L, W_{vb}, P, P_{WT} )$
for $m = 1$ to $C_1$ do
  $h^m = [< START >]$
  # Initial sequence is empty and sequence score is 0
  S.enqueue( $(h^m, 0)$ )
end for

for $i = 1$ to $ML$ do
  # Set correct value of cut-off (no of increments)
  if $i <= 5$ then
    $M = C_1$
  else
    $M = C_2$
  end if

  for $m = 1$ to $M$ do
    # Decoding Step: Only the top $C_1$ $(C_2)$ sequences are retained and expanded
    $(h, s) = S.dequeue()$

    for $l = 1$ to $L$ do
      $h_{add} = h.append(w_l)$
      # Headline synthesis score is calculated here
      $s = P_{WS}(h_{add})$
      $S_f.enqueue((h_{add}, s))$
    end for

    for $j = 1$ to $V$ do
      $h_{add} = h.append(v_j)$
      $s = P_{WS}(h_{add})$
      $S_f.enqueue((h_{add}, s))$
    end for
  end for
  # Set $S$ to $S_f$, and Reset $S_f$ for new search iteration
  $S = S_f$
  $S_f = []$
end for
$Display\_Top\_Headlines( S, T )$
```

Algorithm 2 Procedure to calculate word translation set for news story

```
Procedure: Calculate_Word_Translation_Set
Require: $D$, $L$, $W_{vb} = \{v_1, v_2, ..., v_V\}$, $P = \{p_1, p_2, ...p_L\}$, $P_{WT}$
Ensure: $D_v$

$D_v = \phi$
for $l = 1$ to $L$ do
  # Only for words with verb POS forms
  if $p_l == VB*$ then
    for $i = 1$ to $V$ do
      if $p_{WT}(v_i|w_l) > 0$ then
        # Extract word translation substitution set
        $D_v = D_v \cup v_i$
      end if
    end for
  end if
end for
Return $D_v$
```

Algorithm 3 Procedure to display top T headlines with Scores

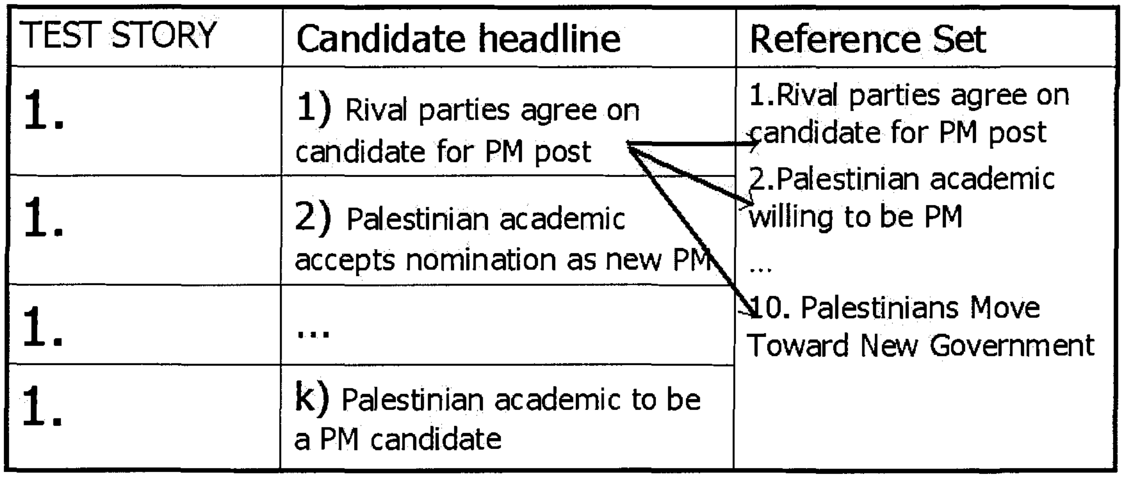
```
Procedure: Display_Top_Headlines
Require: $S$, $T$

# Output Top $T$ Headlines
for $i = 1$ to $T$ do
  $(h, s) = S.dequeue()$
  $print\ h : s$
end for
```

affects generation of headline candidates, since they are used to drive the search space through the space of possible candidate hypothesis. We need a method that explores the parameter space of scaling factors and picks value that maximizes headline generation quality. Maximum Mutual Information (MMI) and Minimum Error Rate training (MER) are two techniques to learn model scaling parameters in the presence of N-best candidate lists. In NLP these techniques find their roots in Statistical Machine Translation (SMT)[25] and [26]. The headline generation setting is analogous to SMT, where source news story and candidate headline translations replace source language sentence and target language candidate translations respectively. Hence it is intuitive and appealing to apply the best model scaling techniques from SMT to the headline generation scheme.

Minimum Error Rate training as introduced by Och [26] uses the N-best list as an approximation to the translation search space, and considers re-scoring the translations with different choices of scaling parameters. This explicitly generates an error surface whose minimum can be inspected, and the corresponding parameter choices reported.

Venugopal et.al. and Och [40][26] show that the space of parameter configurations can be limited to those points that actually cause the error surface to change. This set of points or parameter configurations is termed as *critical set.* The optimization over the entire space of parameter configuration then reduces to optimizing for parameter configuration over a piecewise linear error surface. The configuration (point) which yields the minimum error on the piecewise linear error surface is the optimal configuration.

This setting makes an iterative greedy search strategy through the parameter space quite feasible (optimizing one parameter at a time). A negative corpus level BLEU score measured with respect to a set of candidate reference translations is used as the error surface. Equivalently for headline generation we treat the k-best list at the end of the decoding algorithm as an approximation of the candidate search space and as in SMT, we use negative BLEU score measured with respect to a set of reference headlines as the error surface. Each of headline sequences in the k-best list is matched against all the reference headlines in the reference set to get the individual BLEU scores. The overall BLEU score for the k-best list is obtained by averaging the individual BLEU scores. Fig 4.3 helps understand the use of reference headline set for obtaining BLEU scores. In the figure, the first of the k-best headlines “Rival parties agree on candidate



TEST STORY	Candidate headline	Reference Set
1.	1) Rival parties agree on candidate for PM post	1.Rival parties agree on candidate for PM post 2.Palestinian academic willing to be PM ... 10. Palestinians Move Toward New Government
1.	2) Palestinian academic accepts nomination as new PM	
1.	...	
1.	k) Palestinian academic to be a PM candidate	

Figure 4.3: Use of a reference set to calculate Error score for Minimum Error Rate criterion

for PM post" is matched against all the headlines in the reference set: 1) "Rival parties agree on candidate for PM post" 2) "Palestinian academic willing to be PM" ... 10) "Palestinians Move Toward New Government", and so on for other headlines in the k-best list.

Formally, Minimum Error or Minimum Classification Error (MCE) criterion attempt to minimize the empirical error (as determined by the BLEU evaluation metric) of the k-best decision rule which is explicitly dependent on the values of α. To explicitly model this condition, and borrowing from Och [26] and Venugopal [40], we can define our MCE criterion as below:

$$\begin{aligned} F_{MCE} &= \frac{1}{N}\sum_{i=1}^{N}[Error(h_n^*, R_n)] \\ h_n^* &= \arg\max_{h_n^k \in H_n} \alpha \cdot h(h_n|D) \\ \alpha^* &= \arg\min_{\alpha} F_{MCE} \end{aligned} \tag{4.17}$$

where, $Error(h_n^*, R_n)$ is a function that assigns an error to the selected candidate headline sequence h_n^* with respect to a reference headline set R_n which is available for each news story in the model scaling training data. The decision rule in the second equation above is the same as the decision rule in headline synthesis model discussed earlier.

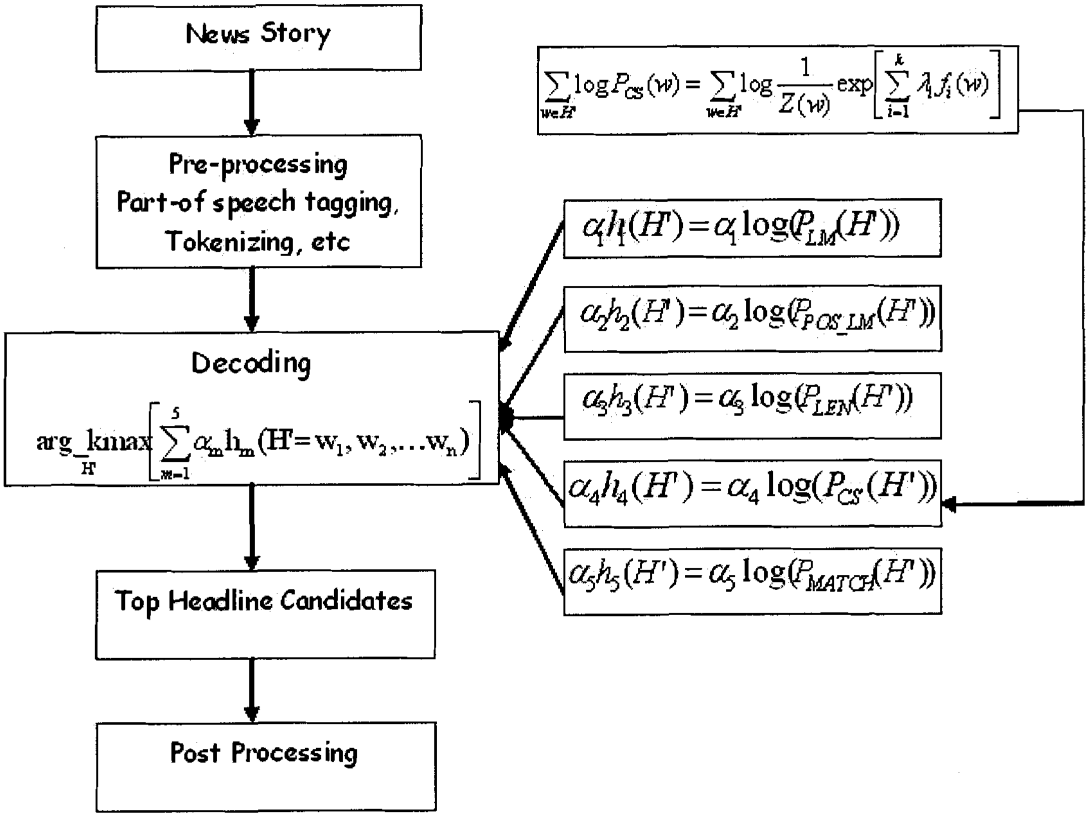

News Story
Pre-processing
Part-of speech tagging,
Tokenizing, etc
Decoding
$\arg\_kmax_{H'} \left[\sum_{m=1}^{5} \alpha_m h_m (H' = w_1, w_2, \ldots w_n) \right]$
Top Headline Candidates
Post Processing
$\sum_{w \in H'} \log P_{CS}(w) = \sum_{w \in H'} \log \frac{1}{Z(w)} \exp \left[\sum_{i=1}^{k} \lambda_i f_i(w) \right]$
$\alpha_1 h_1(H') = \alpha_1 \log(P_{LM}(H'))$
$\alpha_2 h_2(H') = \alpha_2 \log(P_{POS\_LM}(H'))$
$\alpha_3 h_3(H') = \alpha_3 \log(P_{LEN}(H'))$
$\alpha_4 h_4(H') = \alpha_4 \log(P_{CS}(H'))$
$\alpha_5 h_5(H') = \alpha_5 \log(P_{MATCH}(H'))$

Figure 4.4: Architecture of the Headline Generation Framework based on Maximum Entropy Models

Chapter 5

Experiments and Results

Below we present the experimental set-up, tools and packages used, implementation specifics of training, characteristics of the training data, model scaling data and test data and results of our BLEU measure based evaluation on the Google test Data. We also present some sample headlines generated by our framework.

5.1 Experimental Design

The headline news-story pairs used as training data for our system was gathered from two sources. First we gathered $50,000$ news articles and associated headlines from the North American (NA) News Corpus. The NA News corpus is a journalistic text in English from newswire and newspaper sources in US such as the LA Times, Washington Post, New York Times, Reuters, Wall Street Journal, etc. The time periods covered by this collection was from $1994 - 1997$. Additionally, since our test news stories and associated reference set of headlines is gathered from the Google News Service, we extracted approximately 5000 pairs of headlines and news stories by crawling various journalistic websites using the Google News Service. This was done in order to maintain some domain and time period overlap between the training data and test data. Google News service organizes news events into multiple domains such as "World", "US", "Sports", "Entertainment", "Popular". We have restricted the news stories gathered using Google News to just two domains: "World" and "Popular". The combined training data size is approx 55,000 news articles.

Google News clusters multiple news articles related to the same contemporary event together. Table 5.1 for instance shows 5 sample headlines related to the event of "US launching attacks on Somalia". After doing some preliminary analysis we found that a headline A for a news story A in one can be easily used as a headline for news story B in the same cluster and vice versa. In other words, the 5 headlines in Table 5.1 could very well be used as a reference set of headlines for a test story which is not in any of the headline-news article pair A, B, C, D or E.

Headline A: US launches air raids in Somalia
Headline B: Somalia says dozens killed in US attack
Headline C: Many dead after US strike in Somalia
Headline D: US Launches New Attacks in Somalia
Headline E: US strikes terrorist targets in Somalia
... ...

Table 5.1: Sample cluster of headlines for an event on Google News

Additionally, we gather 600 news articles from Google News from the same domains as the training data and distinct from any of the articles or events present in the training data. We do so by extracting 10 headline-news-story pairs each from 60 different news event clusters. The 10 headlines extracted from the same event cluster is used as a reference set for each of the 10 news articles in the same cluster. We maintain the position that a reference set of 5-10 headlines is required to adequately compare system generated headlines against references. This is because unlike summarization where there is a significant overlap between 2 or more human produced reference summaries, headlines can be very different across references because of their very concise nature.

From the 600 news articles we retain 500 news articles for model evaluation (Test data set) and the remaining 100 for tuning model scaling parameters 'α' (development data set) of the headline synthesis model. The characteristics of the 3 data sets are summarized below.

↓ *Attributes\Data Set* →	Training	Development	Test
Size (No of. headline-news-story pairs)	55,000	100	500
Average News Article length	752	516	556
Average Length of Headline	10.82	7.42	7.78
Total tokens (Headlines)	595188	742	3888
Total tokens (News articles)	28425430	51682	278413
Distinct tokens (Headlines)	33803	327	1213
Distinct tokens (News articles)	165778	5177	13627

Table 5.2: Characteristics of Data Sets (Training, development and Test)

5.2 Training Procedure

To help understand the discussion and implementation considerations in model training, we consider the following concrete example. Fig 5.1 shows a sample News Story (Introductory Paragraph) and its accompanying headline.

```
<Title>
Brazilian plane crashes, and 155 are feared dead
</Title>

<Story>
Rescuers on Sunday were clearing thick jungle around the
wreckage of a Brazilian passenger plane that crashed in the Amazon
with 155 people on board, opening the way for teams to begin retrieving
bodies."The chances of finding survivors are increasingly slim,"
Milton Zuanazzi,general director of the National Civil Aviation Agency,
said of what is feared to be the worst aviation disaster in
Brazilian history.
</Story>
```

Figure 5.1: Sample Headline - News Story Pair (Stage 0 - Before Preprocessing)

We divide the training procedure into 1) training to learn the content selection model (λ) and 2) training to learn model scaling parameters (α) using the development set. Prior to beginning corpus level model training, 1) we do some data set cleaning and

```
<Title>
Brazilian/JJ plane/NN crashes/NNS ,/, and/CC 155/CD are/VBP feared/VBN dead/JJ
</Title>

<Story>
Rescuers/NNP on/IN Sunday/NNP were/VBD clearing/VBG thick/JJ
jungle/NN around/IN the/DT wreckage/NN of/IN a/DT Brazilian/JJ
passenger/NN plane/NN that/WDT crashed/VBD in/IN the/DT Amazon/NNP
with/IN 155/CD people/NNS on/IN board/NN ,/, opening/VBG the/DT
way/NN for/IN teams/NNS to/TO begin/VB retrieving/VBG bodies/NNS
./. ‘‘/‘‘ The/DT chances/NNS of/IN finding/VBG survivors/NNS are/VBP
increasingly/RB slim/JJ ,/, ’’/’’ Milton/NNP Zuanazzi/NNP ,/,
general/JJ director/NN of/IN the/DT National/NNP Civil/NNP Aviation/NNP
Agency/NNP ,/, said/VBD of/IN what/WP is/VBZ feared/VBN to/TO be/VB
the/DT worst/JJS aviation/NN disaster/NN in/IN Brazilian/JJ history/NN ./.
</Story>
```

Figure 5.2: Sample Headline - News Story Pair after preprocessing (Stage 0- After Preprocessing)

preprocessing. This includes removing punctuation from the text, Tokenizing both the headline and news story text and Part of Speech tagging the entire training corpus, sentence by sentence as shown in Fig 5.2. 2) we also train auxiliary models and compute statistics which include: The word translation Probability model P_{WT} and the IDF scores for every word in the news story in the training data.

The Content Selection training is carried out in 2 passes over the training data.

- PASS 1: In pass one features are extracted from the news story by gathering context information surrounding story words and identifying whether the word under consideration is present in the headline or not. For Example: The context information for word ‘passenger’ and ‘plane’(Fig 5.3) and the whether it occurs in the headline (outcome), gives the set of features in Table 5.3. At the end of this pass, we remove features whose count lies below a particular cut-off threshold. We use variable feature cut-offs depending on the type of feature and the outcome. Table 5.4 specifies the cut offs used for the content selection feature set.

  For our training data set of 55,000 news stories, the content selection model

```
['1_w=passenger', '2_t=NN', '3_w,w-1=Brazilian,passenger',
 '4_t-1=JJ', '5_t-1,2=DT,JJ', 6_w,w+1=plane,passenger',
 '7_t+1=NN', '8_t+1,2=WDT,NN','9_win=0-10', '10_wti=10-20']
 Outcome: 0

['1_w=plane', '2_t=NN', '3_w,w-1=passenger,plane',
'4_t-1=NN', '5_t-1,2=JJ,NN', '6_w,w+1=thast,plane',
'7_t+1=WDT', '8_t+1,2=VBD,WDT', '9_win=10-20', '10_wti=0-10']
Outcome: 1
```

Figure 5.3: Depiction of context information for 2 story words with outcomes 1 and 0 respectively(Stage 1)

Sample Features
1_w=passenger::outcome=0
2_t=NN::outcome=0
3_w,w-1=Brazilian,passenger::outcome=0
....
1_w=plane::outcome=1
2_t=NN::outcome=1
3_w,w-1=passenger,plane::outcome=1

Table 5.3: Sample Features that fired in the given training example

↓ *Features\Cut-Offs* →	Outcome 1	Outcome 0
Current Story Word	1	2
Word Bi-gram Context	1	2
POS of Current Story Word	2	5
POS Bi-gram of Current Word	2	5
POS Tri-gram of Current Word:	2	5
Word Position in Lead sentence	1	2
Word Position	1	2
First Word Occurrence Position	1	2
Word TF-IDF Range	1	2

Table 5.4: Feature Cutoffs (phase 1)

consists of approximately 900,000 features after dropping features which fail to meet the cut-off criterion.

- PASS 2: In the second pass we gather feature vectors for each story word and outcome pair. The outcome is either inclusion or non inclusion in headline corresponding to 1 and 0 respectively. The feature vector representation for each story word is quite sparse. Only 8-10 features fire for any particular instance of the training tuple.

For learning model parameter weights λ of the content selection (CS) model we use the conditional Maximum Entropy modeling toolkit [46]. The toolkit provides an implementation of the GIS algorithm with smoothing for parameter estimation. A gaussian prior σ^2 with a global variance of 1 is used to regularize the model by seeking a maximum a priori (MAP) solution. The parameter values converge after approximately 50 iterations.

The second part of training involves tuning of model scaling parameters α. The parameters are trained on a development data set of 100 news story-headline pairs using minimum error rate training as outlined in [26][40]. The model scaling vector is initially set to (1,1,1,1,1) where all components contribute equally to the overall Headline Synthesis score. Minimum error rate training is an iterative method that converges to optimal values of model parameters (local minimum of the BLEU measure based error Surface) after 20-30 iterations. While we have used a previous implementation of conditional maximum entropy [46] for training the content selection model, we found it easier to implement minimum error rate training for the model scaling parameters ourselves.

5.3 Results

We conducted experiments faced with the following questions.

- What is the overall BLEU score we achieve on our Google test data after model training?
- Does our model show any improvements over the Naive Bayes Statistical Model?

- What affect does tuning of model scaling parameters (α) has on improving headline generation performance, in other words is there a perceivable difference between using tuned model parameters and using model parameters with equal weights ($\alpha = (1, 1, 1, 1, 1)$)?
- which of the model components ($LM/POS\_LM/LEN/CS/MATCH$) has the most impact on headline generation performance?

5.3.1 Content Selection Results

Content Selection assigns each word in the story a probability of inclusion in the headline. The content selection score is also a good measure of the importance of each word to the story and could be used as a key word selection methodology for a given document.

Figure 5.4 shows the result of applying content selection model over a test news story. The content selection probabilities are indicated in "()" alongside the words. Also the top word selections are listed at the bottom.

5.3.2 BLEU Scores

Table 5.5 and 5.6 show the results of BLEU measure based evaluation on the Google test data on reference data sets of 5 and 10 headlines respectively. For the first table Headline Synthesis model parameters are $\lambda = (1, 1, 1, 1, 1)$ for the models LM, POS_LM, LEN, CS and MATCH respectively. For the second table the models were tuned on a development data set using minimum error rate training. As expected, there are huge performance gains after model tuning is done. Also, the larger reference set gives overall better results as the headlines cover more n-gram matches with the candidates.

In table 5.7 we present the BLEU score comparison between our Maximum Entropy headline generation system and Statistical Naive Bayes (BMW) Headline generation model. Our system outperforms the Naive Bayes approach by a significant margin. This overall improvement can be attributed to our use of considerably better content selection and surface realization techniques that combine rich, complex, overlapping features and knowledge sources into the two models respectively.

```
<Story>
Online gaming firms faced their biggest-ever crisis on Monday
after U.S. Congress passed legislation to end Internet gaming
there, threatening jobs and wiping 3.5 billion pounds
\$6.5 billion off share prices. Britain's PartyGaming,
operator of leading Internet poker site PartyPoker.com,
and rivals Sportingbet and 888 said they would likely pull out of
the United States, their biggest source of revenue, and warned
on future profits. "This development is a significant setback for
our company, our shareholders, our players and our industry,
" PartyGaming Chief Executive Mitch Garber said.

The House of Representatives and Senate unexpectedly
approved a bill early on Saturday that would make it illegal
for banks and credit-card companies to make payments to online
gambling sites. The measure was sent to President George W.
Bush to sign into law, which most analysts see as a certainty.
"We believe that this will have a very material impact on
the long-term prospects of online gambling, and in particular
poker," said analyst Julian Easthope at UBS.
</Story>
<StoryCS>
........
PartyGaming|(0.159999)  Chief|(0.050085)  Executive|(0.061024)
Mitch|(0.070857) Garber|(0.056756)  said|(0.001672)  .|(0.034203)
The|(0.047920)  House|(0.163465) of|(0.116513)  Representatives|(0.057827)
and|(0.027871)  Senate|(0.359593) unexpectedly|(0.004498)
approved|(0.003271)  a|(0.075160)  bill|(0.387927) early|(0.003556)
on|(0.070881)  Saturday|(0.027452)  that|(0.006827)  would|(0.015110)
........
online|(0.063385)  gambling|(0.178380)  sites|(0.033899) .|(0.048326)
The|(0.049862)  measure|(0.086271)  was|(0.004860)  sent|(0.008594)
to|(0.201631)  President|(0.018881)  George|(0.017395)  W.|(0.038068)
Bush|(0.251552) to|(0.156677) sign|(0.021876)  into|(0.012686)  law|(0.179685)
which|(0.000805) most|(0.001418)  analysts|(0.035841)  see|(0.007997)
........
</StoryCS>
<TopSelections>
bill(0.387927)
Senate(0.359593)
Bush(0.251552)
to(0.201810)
law(0.179685)
gambling(0.178380)
....
</TopSelections>
```

Figure 5.4: Result of applying Content Selection Model on Test Data

↓ *Blue Score\Reference Set Size* →	05	10
BLEU 1-gram precision	0.2685	0.3397
BLEU 2-gram precision	0.0706	0.0788
BLEU 3-gram precision	0.0092	0.0217
Overall BLEU Score	0.1222	0.1611

Table 5.5: BLEU Scores on test Data (Without tuning model scaling parameters)

↓ *Blue Score\Reference Set Size* →	05	10
BLEU 1-gram precision	0.3250	0.5604
BLEU 2-gram precision	0.0926	0.1432
BLEU 3-gram precision	0.0242	0.0482
Overall BLEU Score	0.1426	0.2506

Table 5.6: BLEU Scores on test Data (With model scaling parameters tuned on development data set)

↓ *Headline Generation technique\Reference Set Size* →	05	10
Two Stacked Maximum Entropy Model (Without Model Tuning)	0.1222	0.1611
Naive Bayes Model	0.0943	0.1426

Table 5.7: Comparison of BLEU score measures

In table 5.8 we consider the effect on BLEU scores and the impact on performance due to each of the components (LM, LM_POS, LEN, CS, MATCH) of the headline synthesis model. In keeping with our intuition, we find that addition of both LM and POSLM model components and LM in particular significantly boosts CS only based BLEU scores. Addition of the MATCH model component further helps improve the overall BLEU scores on test data. CS, LM, POSLM and MATCH have the most impact on BLEU scores in that order. We also found that LEN model has little to no impact on the overall BLEU scores obtained on test data and hence not included in the table below.

↓ *Model\Reference Set Size* →	05	10
CS	0.0856	0.1077
CS+LM	0.0986	0.1481
CS+LM+POSLM	0.1139	0.1587
CS+LM+POSLM+MATCH	0.1222	0.1611

Table 5.8: Impact of individual Model Components on BLEU Score ($\alpha_i = 1$)

5.3.3 Sample Headline Sequences

In tables 5.9 and 5.10 we look at the overall top headline sequences and the top headline sequences for lengths 1 to 12 for the "Law on Internet Gambling" news story.

Headline Candidate	Score
Bush to sign of	-22.614
Bush to sign bill on	-26.652
Bush to sign of the	-26.835
the House of The Internet gambling	-29.946
The bill of the Internet gambling	-29.982
Bush to end of the Internet gambling	-32.576
Bush to sign bill on the Internet gambling	-35.746
Bush to sign bill on the Internet gambling law	-39.710
Bush to end of the Internet gambling on The Senate bill	-46.988
Bush to sign bill on the Internet gambling site of The law	-50.912

Table 5.9: Top Headlines for "Law on Internet Gambling" news story

Headline Length	Headline Candidate	Score
1	U.S.	-11.661
2	Bush to	-15.360
3	Bush to sign	-19.761
4	Bush to sign of	-22.614
5	Bush to sign bill on	-26.652
6	the House of The Internet gambling	-29.946
7	Bush to end of the Internet gambling	-32.576
8	Bush to sign bill on the Internet gambling	-35.746
9	Bush to sign bill on the Internet gambling law	-39.710
10	Bush to end of the Internet gambling on The law	-43.680
11	Bush to end of the Internet gambling on The Senate bill	-46.988
12	Bush to sign bill on the Internet gambling site of The law	-50.912

Table 5.10: Top Headlines for each length for "Law on Internet Gambling" news story

Chapter 6

Conclusion

Maximum Entropy discriminative models have been very successfully applied to a variety of NLP tasks, hence the thought of using Maximum Entropy models for short summary or headline generation was both intuitive and appealing. In this work we presented a maximum entropy discriminative framework for the headline generation task with three principal components: 1) A Content Selection Model that uses a rich feature set comprising word and POS n-gram features, positional features and word frequency features. 2) A Headline Synthesis Model that combines multiple knowledge sources (models) as feature functions into a second Maximum Entropy model and is used to score candidate headline sequences and 3) A decoding algorithm that uses beam search pruning to explore the headline hypothesis space and generate the optimal headline sentences. Within our framework, choice of headline words is not restricted to only the words present within the story. This is made possible through the use of a word translation model that given sufficient training data learns logical word substitutions for story words.

We also present a novel idea for evaluation of headline generation models/systems that does not rely on human produced references. Our model scaling and evaluation test data sets (reference sets) are extracted using the Google News Service. On reference data sets of size 5 and 10 and in absence of model scaling, our model reports BLEU measure scores of 0.1204 and 0.1587 respectively.

Bibliography

[1] Massih-Reza Amini and Patrick Gallinari. The use of unlabeled data to improve supervised learning for text summarization. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 105–112, New York, NY, USA, 2002. ACM Press.

[2] M. Banko, V. Mittal, and M. Witbrock. Headline generation based on statistical translation. In *the Proceedings of Association for Computational Linguistics*, 2000.

[3] Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22, 1996.

[4] Eugene Charniak. A maximum-entropy inspired parser. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the ACL (NAACL), Seattle, Washington.*, 2000.

[5] J. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:14701480, 1972.

[6] Bonnie Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: a parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop*, pages 1–8, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[7] H. P. Edmundson. New methods in automatic extracting. *Journal of Association for Computing Machinery*, 16, 1969.

[8] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proceedings of the IJCAI*, pages 668–673, 1999.

[9] Jade Goldstein, Mark Kantrowitz, Vibhu O. Mittal, and Jaime G. Carbonell. Summarizing text documents: sentence selection and evaluation metrics. In *the Proceedings of SIGIR 99*, 1999.

[10] P. Kennedy Alexandar G. Hauptmann. Automatic title generation for the informedia multimedia digital library. In *Proceedings of the ACM Digital Libraries, DL-2000, San Antonio, Texas*, 2000.

[11] E. T. Jaynes. Information theory and statistical mechanics. *Statistical Physics, K. Ford (ed.)*, page p181, 1963.

[12] Rong Jin and Alexander G. Hauptmann. Automatic title generation for spoken broadcast news. In *HLT '01: Proceedings of the first international conference on Human language technology research*, pages 1–3, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[13] Rong Jin and Alexander G. Hauptmann. Learning to select good title words: An new approach based on reverse information retrieval. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 242–249, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[14] Rong Jin and Alexander G. Hauptmann. Title generation using a training corpus. In *CICLing '01: Proceedings of the Second International Conference on Computational Linguistics and Intelligent Text Processing*, pages 208–215, London, UK, 2001. Springer-Verlag.

[15] Kevin Knight and Daniel Marcu. Statistics-based summarization - step one: Sentence compression. In *Proceedings of The American Association for Artificial Intelligence Conference (AAAI-2000)*, pages 703–710, 2000.

[16] Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139, 2002.

[17] Reiner Kraft, Farzin Maghoul, and Chang Chi C. Y!q: contextual search at the point of inspiration. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 816–823, 2005.

[18] C. Y. Lin. Cross-domain study of n-gram co-occurrence metrics. In *Proceedings of the Workshop on Machine Translation Evaluation, New Orleans, USA*, 2003.

[19] C. Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out, post-conference workshop of ACL 2004*, pages 311–318, 2004.

[20] H.P. Luhn. The automatic creation of literature abstracts. *I.B.M. Journal of Research and Development*, 2, 1958.

[21] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *COLING-02: Proceedings of the 6th conference on Natural language learning*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[22] Inderjeet Mani and Mark T. Maybury. *Advances in Automatic Text Summarization.* MIT Press, Cambridge, MA, USA, 1999.

[23] Daniel Marcu. From discourse structures to text summaries. In *the Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, Madrid, Spain*, pages 82–88, 1997.

[24] R. McDonald. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of the EACL'06*, 2006.

[25] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[26] Franz Josef Och. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[27] C.D. Paice and P.A. Jones. The identification of important concepts in highly structured technical papers. In *the Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 69–78, 1993.

[28] C.D. Paice. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26, 1990.

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[30] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Empirical Methods in Natural Language Conference,*, 1996.

[31] A. Ratnaparkhi, S. Roukos, and R. Ward. A maximum entropy model for parsing. In *Proceedings of the International Conference on Spoken Language Processing*, pages 803–806, 1994.

[32] A. Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1998.

[33] U. Reimer and U. Hahn. Text condensation as knowledge base abstraction. In *the Proceedings of Fourth Conference on Artificial Intelligence Applications*, pages 338–344, 1988.

[34] Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19, 1997.

[35] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.

[36] G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summary. *Information Process And Management,*, 33, 1997.

[37] E.F. Skorokhodko. Adaptive method of automatic abstracting and indexing. In *the IFIP Congress, Ljubljana, Yugoslavia*, pages 1179–1182, 1972.

[38] Christoph Tillmann and Hermann Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133, 2003.

[39] P.D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2, 2000.

[40] Ashish Venugopal and Stephan Vogel. Considerations in maximum mutual information and minimum classification error training for statistical machine translation. In *the Proceedings of EAMT-05*, 2005.

[41] Ruichao Wang, Nicola Stokes, William P. Doran, Eamonn Newman, Joe Carthy, and John Dunnion. Comparing topiary-style approaches to headline generation. In *Proceedings of the 27th European Conference on Information Retrieval (ECIR-05)*, 2000.

[42] David Zajic, Bonnie Dorr, and Richard Schwartz. Headline generation for written and broadcast news, lamp-tr-120, cs-tr-4698,. Technical report, College Park, Maryland, USA, 2005.

[43] David Zajic and Bonnie Dorr. Bbn/umd at duc2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston, USA*, page 112119, 2004.

[44] http://news.Google.com.

[45] http://wordnet.princeton.edu.

[46] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.

[47] http://nlp.stanford.edu/software/index.shtml.