Automatic Headline Generation for Newspaper Stories

David Zajic & Bonnie Dorr
University of Maryland
{dmzajic,bonnie}@umiacs.umd.edu
Richard Schwartz, BBN
schwartz@bbn.com

Abstract: In this paper we propose a novel application of Hidden Markov Models to automatic generation of informative headlines for English texts. We propose four decoding parameters to make the headlines appear more like Headlinese, the language of informative newspaper headlines. We also allow for morphological variation in words between headline and story English. Informal and formal evaluations indicate that our approach produces informative headlines, mimicking a Headlinese style generated by humans.

1. Introduction

We are investigating the task of automatic generation of headlines for news stories. Our focus is currently on headline generation for English texts. Although news stories already have human-generated headlines, these pre-existing "abstracts" are frequently not descriptive enough for our purposes, particularly in the case of eye-catchers (e.g., "Only the Strong Survive") or even in the case of *indicative* abstracts (e.g., "French Open").

In contrast to human-generated newspaper headlines, our approach produces *informative* abstracts, describing the main theme or event of the newspaper article (e.g., "Wide Gap Between 3 Best Players and Others at French Open"). The long-term goal of our effort is to apply our approach to noisy input, e.g., multilingual text and speech broadcasts, where the application is clearer, as these inputs don't have viable (or any) human-generated headlines.

Other researchers have investigated the topic of automatic generation of abstracts, but the focus has been different, e.g., sentence extraction (Edmundson, 1969; Johnson et al, (1993; Kupiec et al., 1995; Mann et al., 1992; Teufel and Moens, 1997; Zechner, 1995), processing of structured templates (Paice and Jones, 1993), one-sentence-at-a-time compression (Knight and Marcu, 2001; Luhn, 1958), and generation of abstracts from multiple sources (Radev and McKeown, 1998). We focus instead on the construction of headline-style abstracts from a single *story*.

Our method is to form headlines by selecting headline words from story words found in the newspaper article. As a first approximation, we select headline words from story words in the order that they appear in the story. In addition, morphological variants of story words may appear as headline words.

Consider the following excerpt from a news story:

(1) <u>Story Words:</u> After months of debate following the Sept. 11 terrorist hijackings, the Transportation Department has decided that airline **pilots** will **not** be **allowed to have guns in** the **cockpits**.

<u>Generated Headline:</u> Pilots not allowed to have guns in cockpits

In this case, the words in bold form a fluent and accurate headline for the story. However, it is often necessary to use a morphological variant of a story word to form a fluent headline. For example, headlines are usually written in present tense, while stories are written in past tense:

(2) <u>Story Words</u>*:* President **Bush**, in a speech harshly critical of Fidel Castro, **said** today that **he would not lift** a trade

embargo against Cuba without substantial movement toward democracy there.

Generated Headline*:* Bush *says* he *will* not lift embargo against Cuba.

(3) Story Words*:* The **Civic Center doesn't** come close to **meeting** current earthquake **safety standards**.

Generated Headline*:* Civic Center doesn't *meet* safety standards.

In both (2) and (3), the story words in boldface form accurate headlines. However, it is preferable to use the morphological variants shown in italics in the corresponding headlines. In (2), the morphological variants are used to convert the headline into the more usual present tense of Headlinese (Mårdh, 1980). In (3), a morphological variant is used to make the headline grammatical.

In this paper, we present our technique for producing headlines using a Hidden Markov Model (HMM). We first discuss the results of our feasibility testing—illustrating that our approach is a promising path to follow. Next, we describe the application of HMM to the problem of headline generation. After this, we discuss the decoding parameters we use to produce results that are more headline-like. We then present our morphological extensions to the basic headline-generation model. Finally, we discuss two evaluations—one by human and one by machine—for assessing the coverage and general utility of our approach to automatic generation of headlines.

2. Feasibility Testing

To determine the feasibility of our headline-generation approach, we first attempted to apply our "select-words-in-order" technique by hand. We examined 56 stories randomly chosen from the Tipster corpus. Taking hand-selected story words in the order in which they appeared, we were able to construct fluent and accurate headlines for 53 of the stories. The remaining 3 stories were a list of commodity prices, a chronology of events, and a list of entertainment events. We conclude that our approach has promise for stories that are written as paragraphs of prose.

As part of this initial feasibility evaluation, we observed that only 7 of our 53 headlines used words beyond the 60th story word, and of those only one went beyond the 200th word. Stories whose headlines required the later words tended to be human-interest stories with attention-grabbing introductions or they appeared to be excerpts from the middle of larger stories. Thus, in our current model, we adopt the additional constraint that story words must be chosen from the first N words of the story, where N has been intuitively set at 60.

3. Approach: Noisy Channel Model

Our algorithm for selecting story words to form headlines is based on a standard NoisyChannel Model of processing—with a subsequent decoder for producing headline words from stories. The Noisy Channel approach has been used for a wide range of natural language processing (NLP) applications including speech recognition (Bahl et al. 1983), machine translation (Brown et al.1990), sentence boundary detection (Gotoh and Reynolds 2000), spelling correction (Mays et al. 1990), language identification (Dunning 1994), part-of-speech tagging (Cutting et al.1992), syntactic parsing (Collins 1997b; Charniak 1997), semantic clustering (Lin 1998; Pereira et al. 1993), sentence generation (Langkilde and Knight 1998; Bangalore and Rambow 2000), and summarization (Knight 2000). We adopt a similar technique to that of each of these applications, but we apply it to a new domain: generation of headlines from stories.

The intuition is to treat stories and headlines as the joint output of a generative model. Our approach is to find the headline most likely to have been generated jointly with a given story. In a given story, some words will be identified as headline words. The headline will be composed of the headline words, or morphological variants of the headline words. Thus, stories consist of headline words (or morphological variants of headline words) with many other words interspersed amongst them, and the most likely headline is determined by calculating the most likely set of headline words given that the observed story was generated.

Formally, if H is a ordered subset of the first N words of story S, we want to find the H which maximizes the likelihood that H is the set of headline words in story S, or:

$$\arg\max_H P(H \mid S)$$

It is difficult to estimate P(H|S), but this probability can be expressed in terms of other probabilities that are easier to compute, using Bayes' rule:

$$P(H \mid S) = \frac{P(H)P(S \mid H)}{P(S)}$$

Since the goal is to maximize this expression over H, and P(S) is a constant with respect to H, P(S) can be omitted. Thus we wish to find:

$$\arg\max_H P(H)P(S \mid H)$$

3.1 Source Model: Bigram Estimates of Headline Probabilities

We estimate *P(H)* using the bigram probabilities of the headline words used in the story:

$$P(H) = P(h_1 \mid start)P(h_2 \mid h_1)...P(h_n \mid end)$$

3.2 Generative Model: Using HMMs for Story Generation from Headlines

To estimate *P(S|H)* we must consider the process by which a story is generated. This process can be represented as a Hidden Markov Model (HMM). A HMM is a weighted finite-state automaton in which each state probabilistically emits a string. The simplest HMM to generate stories with headlines is shown in Figure 1.

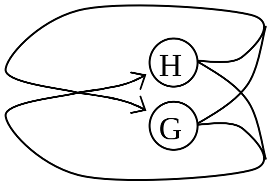

Figure 1

Consider the story in (1). The H state will emit the words in bold (pilots, not, allowed, to, have, guns, in, cockpits), and the G state will emit all the other words. The HMM will transition between the H and G states as needed to generate the words of the story.

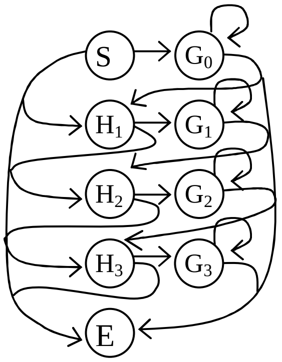

Figure 2

We use a unigram model of stories and a bigram model of headlines based on a corpus of 496215 stories from Associated Press, Wall Street Journal and San Jose Mercury News. Because of the bigram model of the headline language, the HMM in Figure 1 will not be sufficient. The HMM for a three-word story is shown in Figure 2 above. There should be an H state in the HMM for each word in the headline vocabulary. Since we can observe the words in the story, it is sufficient to have an H state for each word in the story. Each H state will have a corresponding G state which emits story words until the next headline word and remembers the previous emitted headline word.

The HMM starts in start state S. It can transition from S to any H state or to state G_0. When the HMM is in an H state it emits a headline word. From an H state, the HMM can transition to any later H state or to the corresponding G state. From any G state, the HMM can stay in that state or transition to any later H state. Any state can transition to the end state E.

Suppose we observe the story in (1) as the output of an HMM. There are 28 words in that story, so there will be 28 H states, 29 G states, a start state S and an end state E in the HMM. The headline in (1) can generate the story as follows. The HMM will start in state S, emit a start symbol and transition to state G_0. It will stay in G_0 and emit the words *after, months, of, debate, ..., decided, that* and *airline.* Then it will transition to state H_{pilots} and emit the word *pilots.*

The next word in the story is not a headline word, so the HMM transitions to the corresponding G state, G_{pilots}, which emits *will.* Note that being in state G_{pilots} allows the machine to remember that *pilots* is the last emitted headline word. The next story word is a headline word, so we transition to H_{not} and emit *not*. Skipping ahead to after $H_{allowed}$ has emitted *allowed*, we note that the next story word is also a headline word. In this case, the HMM does not go into the corresponding G state, but instead goes directly to $H_{to.}$

Finally, after *cockpits* is emitted by $H_{cockpits}$, the HMM goes to the end state. If there had been more words in the story after cockpits, they would all be emitted by $G_{cockpits}$, then the HMM would go to the end state.

Transitions from an H state to a later H state corresponds to a *clump* of sequential headline words in the story. A transition from an H state to a G state corresponds to the end of a clump and the start of a *gap*, i.e., a headline word followed by non-headline word.

Conversely, a transition from a G state to a H state corresponds to the end of a gap and the start of a clump.

This process can be thought of as one in which a story and a headline are generated simultaneously. Alternatively, we can think of the headline as the input to an HMM controlling the sequence of H states, but in which the model is free to transition to G states at any time. This view fits the Noisy Channel Model interpretation.

P(S|H) is estimated using this HMM. The H states can emit only their specific word from the headline vocabulary with probability 1. The G states can emit any word w in the general language vocabulary with probability *P(w).*

Every possible headline corresponds to a path through the HMM which successfully emits the story. The path through the HMM described above is not the only one that could generate the story in (1). Other possibilities are:

(4) Transportation Department decided airline
pilots not to have guns

(5) Months of the terrorist has to have cockpits

Although (4) and (5) are possible headlines for (1), the conditional probability of (5) given (1) will be lower than the conditional probability of (4) given (1).

3.3 Viterbi Decoding

We use the Viterbi algorithm to select the most likely headline for a story. The implementation takes advantage of the constraints that we imposed on headlines: that headline words are taken from the story in the order that they appear. Headline states can only emit a specific word, and all other words have zero probability. Each headline state has transitions only to the following headline state or to the corresponding G state.

4. Decoding Parameters

In the course of our investigation, we added four decoding parameters motivated by intuitive observations of the output. Our goal was to make the results more like Headlinese. The decoding parameters are: (1) a length penalty, (2) a position penalty, (3) a string penalty and (4) a gap penalty. Note that the incorporation of these parameters changes the values in the cells from log probabilities to relative desirability scores.

We tested different values of the four parameters by trial and error. A logical extension to this work would be to attempt to learn the best setting of these parameters, e.g., through Expectation Maximization (Collins, 1997a).

4.1 Length Penalty

The most salient parameter is the length penalty. We have observed that headlines are usually 5 to 15 words long. The initial translation model had no pressure for headlines in this length range. It is possible for the algorithm to generate headlines of length N which include all the story words, or of length zero.

The length penalty biases the algorithm towards shorter or longer headlines as follows. The transition probability from a G state to itself is multiplied by the length penalty. A length penalty greater than one will favor paths which spend more time in G states, and thus have fewer headline words. A length penalty less than one will favor paths which spend less time in G states, and thus have more headline words. The goal is to nudge the headline length into a specific length range, so no single length penalty is suitable for every story. We iterate the Viterbi algorithm, adjusting the length penalty until the headline length falls in the desired range.

4.2 Position Penalty

We observed that, in the human-constructed headlines, the headline words tended to appear near the front of the story. The position penalty is used to favor headline words that occur early in the story. The story word in the n^{th} position is assigned a position penalty of p^n, where p is a positive number less than one. The emission probabilities on H states are multiplied by the position penalty for the position of the word being considered. Thus words near the front of the story carry less of a position penalty than words farther along.

This technique often fails in the case of human interest and sports stories that start with a hook to get the reader’s attention, before getting to the main topic of the story.

4.3 String Penalty

We observed that the human-constructed headlines often contained contiguous strings of story words in the headlines. Examples (1) and (2) above illustrate this with strings such as “allowed to have guns,” and “embargo against Cuba.” The string penalty is used as a bias for “clumpiness”, i.e., the tendency to generate headlines composed of strings of contiguous story words. Each transition from an H state to its G state is multiplied by the string penalty. A string penalty lower than one will cause the algorithm to prefer clumpy headlines.

4.4 Gap Penalty

Very large gaps between headline words tend to be a sign of great effort from the human to piece together a headline from unrelated words. We believe that the algorithm would not be nearly as successful as the humans in constructing large gap headlines, and that allowing it to try would cause it to miss easy, non-gappy headlines.

The gap penalty is used to bias against headline gappiness, i.e., the tendency to generate headlines in which contiguous headline words correspond to widely separated story words. At each transition from a G state to a H state, a gap penalty is applied which depends on the size of the gap since the last headline word was emitted. This can also be seen as a penalty for spending too much time in one G state. Low gap penalties will cause the algorithm to favor headlines with few large gaps.

5. Morphological Extensions to the Model

Headlines usually use verbs in the present tense while stories use verbs in the past tense. This observation suggests that our initial algorithm omitted important content words from headlines because their probability in the headline language model is low.

The algorithm has been modified to accommodate morphological variations as follows. Each story word is expanded into its set of morphological variants, such that each story-position is associated with a set of strings. In the HMM there is a H state and a G state for each story-position string pair. For example, if the second word in the story is “said”, there can be an H state capable of emitting “says,” and the word “says” can appear in the generated headline.

The emission probability for an H state is nonzero for all the morphological variants of that word. At present this probability is 1/n, where n is the number of morphological variants. In future work, this will be biased in favor of the morphological variations that are observed between headlines and stories.

6. Evaluation

We conducted two evaluations—one informal (a human assessment) and one formal (an automatic evaluation using Bleu).

6.1 Human Assessment

An informal evaluation was done in which the authors evaluated the headlines generated for 30 stories for fluency and accuracy as the decoding parameters and morphological variants were incorporated into the algorithm. The headlines were scored subjectively from 1 to 5 for fluency and accuracy. The average scores for each added parameter are shown in the table.

Experiment	Average Fluency	Average Accuracy
Base	1.17	1.86
Limit Len	2.03	1.73
Pos Penalty	2.30	2.23
Str Penalty	3.13	2.53
Gap Penalty	3.73	3.10
Morph Var	3.57	3.03

These informal results indicate that our intuitively determined parameters do have a positive impact on the results, and bear further study. The slight drop in performance when morphological variants are added seems to be due to failures in noun-verb agreement, which is not surprising in a bigram language model. The morphological variants might provide more benefit when they are trained to reflect actual differences between headlines and stories

6.2 Bleu: Automatic Evaluation

We ran an automatic Bleu-style evaluation of our results, taking three sets of 62 human reference headlines as our basis for comparison.[1] The humans generating the reference headlines were asked to produce headlines by selecting words from the story, in order, with morphological variants allowed—i.e., an *informative* abstract, the type of headline we strive to achieve. Note that by instructing the humans to follow the rules of our general method, the human reference headlines are constrained to headlines that the system could possibly produce. We ran Bleu comparing our headline output to these reference translations.

We found that the results corresponded to our expected degree of improvement for each of the headline generation models, as shown in the table below.

Experiment	Bleu Result
Base	.1475
Limit Len	.1886
Pos Penalty	.2426
Str Penalty	.2863
Gap Penalty	.2971
Morph Var	.3104

Note in particular that in the automatic evaluation, the model incorporating morphological variants more closely corresponds to that of the human reference headlines (i.e., Headlinese), indicating that our hypothesis in Section 6.1 above may be correct: the model that incorporates morphological variants may be more beneficial when trained to reflect Headlinese.

In additional experimentation, we computed the Bleu score of each human against the other two. Taking the best of these scores (.4231) to be an upper bound, we find that the Bleu evaluation of our machine-generated headlines approaches this upper bound.

We also computed the Bleu score of our machine-generated headlines with respect to the original newspaper headlines (.1079). Taking this score to be a baseline, we find that our system scored significantly higher, even without the decoding parameters. This is an expected result, since our headlines are intended to be informative, whereas the original headline is frequently an eye-catcher.

Finally, a comparison of the human-generated headlines against the original textual headlines produces an average score of .0566 across the three reference headlines. This indicates that the human reference set is even less similar to the original text headlines than our machine-generated headlines. This last observation points to an interesting area for future investigation in which we restrict our training set to those naturally occurring headlines whose structure matches the constraints we have imposed on our machine-generated headlines, thus more closely mimicking the behavior of our human references.

7. Future Work

We plan to refine the settings of the decoding parameters through expectation maximization. We will improve our language models by using trigrams for headlines and bigrams for stories. In addition, we will test our system in a setting where we have trained it on only those naturally occurring headlines that fit the constraints described above. Other goals include removing the order constraint on our

[1] For a cogent description of Bleu (BiLingual Evaluation Understudy), see (Papineni et al., 2001).

machine-generated headlines, automatic recognition of stories for which this approach is not suitable (e.g. lists, calendars, formatted data), and cross-linguistic headline-generation.

Acknowledgements

This work has been supported, in part, by BBN Contract 020124-7157 and Darpa/TIDES Contract N66001-00-2-8910. We would like to thank Nizar Habash and Stephen Martin for generating "reference headlines" for 62 stories.

Bibliography

Bahl, L.R., Jelinek, F., and Mercer, R.L. (1983). A maximum likelihood approach to speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179-190

Bangalore S. and Rambow, O. (2000) "Exploiting a probabilistic hierarchical model for generation," In *Proceedings of COLING* 2000.

Brown, P.F., Cocke, J. Pietra, S.A.D., Pietra, V.J.D., Jelinek, F., Lafferty, J.D., Mercer, R.L., and Roossin, P.S. (1990). A statistical approach to machine translation, *Computational Linguistics*, 16(2):79-85

Charniak, E. (1997) Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI-97*.

Collins, M (1997a), "The EM Algorithm (In fulfillment of the Written Preliminary Exam II requirement)" 1997.

Collins, M. (1997b). Three generative lexicalised models for statistical parsing. In *Proceedings of the 35th ACL*, 1997.

Cutting, D, Kupic, J., Pedersen, J. and Sibun, P. (1992). A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, 1992. ACL.

Dunning, T. (1994). Statistical identification of language. Technical Report MCCS 94-273, New Mexico State University.

Edmundson, H. (1969). "New methods in automatic extracting." *Journal of the ACM*, 16(2).

Gotoh, Y. and Reynolds, S. (2000). Sentence boundary detection in broadcast speech transcripts, Proceedings of the International Speech Communication Association (ISCA) Workshop: Automatic Speech Recognition: Challenges for the New Millenium (ASR-2000), Paris, September 2000.

Johnson, F. C., Paice, C. D., Black, W. J., and Neal, A. P. (1993). "The application of linguistic processing to automatic abstract generation." *Journal of Document and Text Management*, 1(3):215-42.

Knight, Kevin and Daniel Marcu (2001). "Statistics-Based Summarization Step One: Sentence Compression (2000)," In *Proceedings of AAAI-2001*.

Kupiec, J., Pedersen, J., and Chen, F. (1995). "A trainable document summarizer." In *Proceedings of the 18th ACM-SIGIR Conference*.

Langkilde, I. and Knight, K. (1998) Generation that exploits corpus-based statistical knowledge," In *Proceedings of COLING-ACL*, 1998

Lin, D. (1998) Automatic retrieval and clustering of similar words. In *Proceedings of Coling/ACL-98*, 1998.

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

Mårdh, I. (1980). Headlinese: On the Grammar of English Front Page Headlines, Malmo.

Mays, E., Damerau, F.J. and Mercer, R.L. (1990). Context-based spelling correction. In *Proceedings, IBM Natural Language ITL,* France, 517-522

Paice, C. D. and Jones, A. P. (1993). “The identification of important concepts in highly structured technical papers.” In *Proceedings of the Sixteenth Annual International ACM SIGIR conference on research and development in IR*.

Papineni, Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu (2001). “IBM Research Report Bleu: a Method for Automatic Evaluation of Machine Translation,” IBM Research Division Technical Report, RC22176 (W0109-022), Yorktown Heights, New York.

Pereira F., Tishby N., and Lee, L. (1993) Distributional Clustering of English Words. In *Proc of 31st ACL*, 1993.

Radev, Dragomir R. and Kathleen R. McKeown (1998). “Generating Natural Language Summaries from Multiple On-Line Sources.” *Computational Linguistics*, 24(3):469--500, September 1998.

Teufel, Simone and Marc Moens (1997). “Sentence extraction as a classification task,” In *Proceedings of the Workshop on Intelligent and scalable Text summarization*, ACL/EACL-1997, Madrid, Spain.

Zechner, K. (1995). “Automatic text abstracting by selecting relevant passages.” Master's thesis, Centre for Cognitive Science, University of Edinburgh.