Headline Generation using a Training Corpus

Rong Jin and Alexander G. Hauptmann

Language Technology Institute, School of computer science, Carnegie Mellon University,
5000 Forbes Ave., Pittsburgh, PA15213, U.S.A
{rong+,alex+}@cs.cmu.edu

Abstract. This paper discusses fundamental issues involved in word selection for title generation. We review several common methods that have been used for title generation and compare the performance of those methods using an F1 metric. Both a KNN (k nearest neighbor) method, which we are the first to apply to title generation, and a limited-vocabulary Naïve Bayesian method outperform other evaluated methods with an F1 score of over 20%. We conclude that KNN (k nearest neighbor) is a simple and promising method in title generation under the assumption that strong content overlap exists between the training corpus and the test collection. We also point out ways to improve the performance both from the learning side and from the generation side.

1 Introduction

To create a title for a document is to engage in a complex task: One has to understand what the document is about, one has to know what is characteristic of this document with respect to other documents, one has to know how a good title sounds to catch attention and how to distill the essence of the document into a title of just a few words. For research, the title generation problem is very attractive because it produces a very compact representation of the original document, which will help people to understand the important information contained in the document quickly. Title generation is also a very difficult problem from the viewpoint of machine learning and natural language processing.

In our approach to the title generation problem we will assume the following:

First, the system will be given a set of training data. Each datum consists of a document and corresponding title. After exposure to the training corpus, the system should be able to generate a title for any unseen document.

We decompose the title generation problem into two parts: learning and analysis from the training corpus and generating a sequence of title words to form the title. For the learning part, we have to decide which parts of knowledge the system needs to learn and how to represent the knowledge learned from the training data. There are four pieces of knowledge that can be induced from training data:

1. Knowledge from the analysis of the document to be titled (referred to as Kd),
2. Knowledge from the analysis of the documents in the training corpus (we will refer to this as the language model for all the documents in the training corpus, or LD),
3. Knowledge from the analysis of the titles in the training corpus (referred to as the language model for all the titles in the training corpus, or LT),
4. Knowledge from analysis of the correlation between documents and their corresponding titles (referred to as the joint document/title language model for the training corpus, or JL).

From the viewpoint of generating part, we decompose the issues involved as follows:

1. Choosing appropriate title words,
2. Deciding how many title words are appropriate for this document title,
3. Finding the correct sequence of title words that forms a readable title ‘sentence’.

Historically, the title generation task is strongly connected to traditional summarization [2,3] because it can be thought of extremely short summarization. Traditional summarization has emphasized the extractive approach, using selected sentences or paragraphs from the document to provide a summary [4,5,6]. The weakness of this approach is that most of knowledge (referred to as LD, LT and JL above) embedded in the training corpus is ignored.

More recently, some researchers have moved toward “learning approaches” that take advantage of training data. Witbrock and Mittal have tried a form of. They ignore all document words that are not in the title language model LT. Only document words that effectively reappear in the title of a document are counted when they estimate the probability of a title word w_t given a document word w_d as: $P(w_t|w_d)$ where $w_t = w_d$. While the Witbrock/Mittal Naïve Bayesian approach is not in principle limited to this constraint, our experiments show that it is a very useful restriction. Kennedy tried a generative approach with an iterative Expectation-Maximization algorithm using most of the document vocabulary [13].

For the purpose of comparison over a test pool and to present contrastive results, in this paper we explore the following learning approaches:

Extractive summarization which selects the “best” sentence from the document as a title. This approach uses a term frequency inverse document frequency (tf.idf) based approach to weighting individual sentences.

Naïve Bayesian approach with limited vocabulary. This closely mirrors the experiments reported in [1].

KNN (k nearest neighbors) which treats title generation as a special classification problem. We consider the titles in the training corpus as a fixed set of labels, then the task of generating a title for a new document is essentially the same as selecting an appropriate label (title) from the fixed set of training labels. The task reduces to finding the document in the training corpus, which is most similar to the current document to be titled. Standard document similarity vectors can be used. The new document title will be set to the title for the training document that is most similar to the current new document. One drawback of this approach is the assumption that any new document will be appropriately titled by an existing title in the training dataset, which is fixed. This approach makes no attempt to explore LD and LT knowledge and only makes use of small piece of knowledge JL.

Naïve Bayesian approach with full vocabulary. In this approach, we compute the probability of a title word given a document word for all words in the training data, not just those that where $w_t = w_d$. As a result, the search space becomes much larger, but the principles are the same as in [1].

The outline of this paper is as follows: Section 1 gave an introduction to the title generation problem. The details of the experiment and analysis of results are presented in Section 2. Section 3 discusses our conclusions drawn from the experiment and suggests possible improvements.

2 The Contrastive Title Generation Experiment

In this section we describe the experiment and present the results. Section 2.1 describes the data. Section 2.2 discusses the evaluation method. Section 2.3 gives a detailed description of all the methods, which were compared. Results and analysis are presented in section

2.1 Data Description

The experimental dataset comes from a CD of 1997 broadcast news transcriptions published by Primary Source Media [14]. There were a total of 50,000 documents and corresponding titles in the dataset. The training dataset was formed by randomly picking four documents-title pairs from every five pairs in the original dataset and the leftover one pair was put into the held-out test collection. The size of training corpus was therefore 40,000 documents and their titles and the size of test collection was 10,000. By separating training data and test data in this way, we ensure strong overlap in topic content between training dataset and test dataset, which gives the learning algorithm a better chance to play a big role in the final performance.

2.2 Evaluation

In this paper, evaluate title generation using the F1 metric[9]. For an automatically generated title T_{auto}, F1 is measured against correspondent human assigned title T_{human} as follows:

$$F1 = 2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall}) \quad \textbf{(1)}$$

Here, precision and recall is measured as the number of identical words in T_{auto} and T_{human} over the number of words in T_{auto} and the number of words in T_{human} respectively. Obviously the sequential word order of the generated title words is ignored by this metric. There are several other metrics that take this ordering information into consideration: string edit distance (DTW) [12] and maximal sub-string [1]. Since we are focusing here on the choice of appropriate title words, F1 is the most appropriate measure for this purpose. Furthermore, we ignore those parts of each title generation approach which orders the generated title words into a ‘sentence’. To make it fair for each approach, all approaches only generate 6 title words, which is the average number of title words in the training corpus. Stop words are removed throughout the training and testing documents and titles.

2.3 Description of the Compared Approaches

The symbols used throughout the paper are defined in Table 1.

Symbol	Definition
doc_tf(w, i)	Term frequency of word w in i-th document
Doc_tf(w)	Term frequency of word w in the new document
title_tf(w, i)	Term frequency of word w in i-th title
N	Number of document-title pairs in training dataset
max_tf	Maximum term frequency in a document
Tf	Term frequency of a word in a document
Df	The number of documents that a word occurs in a collection

Table 1. Symbols used throughout the paper

2.3.1 Naïve Bayesian Title Generation with limited vocabulary (NBL). Essentially, this algorithm duplicates the work in [1], which tries to capture the correlation between the words in the document and the words in the title. The following steps are performed:

1. *Learning step*: From the training corpus, count the occurrence of each document-word-title-word pair whose document word and title word is the same. C_w represents the occurrence of such a pair when both document word and title word are w. C_w can be expressed as following:

$$C_w = \Sigma_{j=1}^{N} \text{doc\_tf}(w, j) \times \text{title\_tf}(w, j) \qquad (2)$$

   The sum goes over all document-title pairs in the training corpus. The conditional probability P(titleword w | documentword w) is obtained by dividing C_w by $\Sigma_{j=1}^{N}$ doc_tf(w, j)

2. *Generation step*: To generate a title for a new document, we compute the generating potential G_w for each word w in the new document:

$$G_w = \text{doc\_tf}(w) \times P(\text{titleword } w \mid \text{documentword } w\,) \qquad (3)$$

   Here, doc_tf(w) is the term frequency of word w in the new document. Those title words with largest generating potential G_w will be chosen to form the title.

2.3.2 Naïve Bayesian approach with full vocabulary (NBF). In previous approach, we count only the cases where the title word and the document word are same. This restriction is based on the assumption that a document word is only able to generate a title word with same surface string. The constraint can be easily relaxed by counting all the document-word-title-word pairs. The details are as follows:

1. *Learning step*: Use C(dw, tw) to represent the occurrence of document-word-title-word pairs when the document word is dw and the title word is tw. C(dw, tw) can be expressed as:

$$C(dw, tw) = \Sigma_{j=1}^{N} doc\_tf(dw, j) \times title\_tf(tw, j) \quad \textbf{(4)}$$

   The sum goes over all the document-title pairs in the training dataset. Then we normalize C(dw, tw) to C_norm(dw, tw) as follows:

$$C\_norm(dw, tw) = C(dw, tw) / \Sigma_{tw} C(dw, tw) \quad \textbf{(5)}$$

   This normalisation is necessary because we treat C_norm(dw, tw) as a conditional probability and the property of a probability demands the normalization. A detailed explanation can be found in [7].

2. *Generation step*: To generate a title for a new document, we compute the generating potential G_{tw} for each possible title word tw in the new document:

$$G_{tw} = \Sigma_{dw} doc\_tf(dw) \times C\_norm(dw, tw) \quad \textbf{(6)}$$

   Here, doc_tf(dw) is the term frequency of word dw in the new document and the sum in the above equation goes over all the words in the new document. Those title words with maximum generation potential G_{tw} will be chosen for the title. This full vocabulary version of Naïve Bayesian can be compared directly with the limited vocabulary version to see the importance of vocabulary restriction.

2.3.3 Extractive Summarization Approach using TF/IDF. We have mentioned the similarity between title generation and story summarization. In this paper we include a sentence-based summarization method [3] for comparison. The extraction of sentence is based on tf.idf information. The detail is following:

1. *Learning step*: Compute the idf values for all words in the documents of the training corpus. Ignore the titles.
2. *Generating step*: Decompose the new document into sentences. Compute the average tf.idf value for each sentence. If an unknown word occurs in a sentence, assign zero to the idf of that word. Use the sentence with highest average tf.idf as the title.

2.3.4 K Nearest Neighbor approach (KNN). This algorithm is similar to the KNN algorithm applied to topic classification in [8]. It treats the titles in the training corpus as a set of fixed labels. For each new document, instead of creating new title, it tries to find an appropriate "label", which is equivalent to searching the training document set for the closest related document. This training document title is then used for the new document. The algorithm proceeds as follows:

1. *Learning step*: Index the documents in the training corpus using the SMART document retrieval system [10] using the standard stop word list. The SMART document weighting schema used was "ATC", which is:
   A --- (tf +0.5)/(max_tf+1.0)
   T --- log(collection_size/df)
   C --- Euclidian vector length normalization
2. *Generating step*: For each new document, compute its similarity to all the documents in the training corpus by SMART and output the title of the document in the training corpus most similar to the new document as the title for the new document.

2.4 Results and Observations.

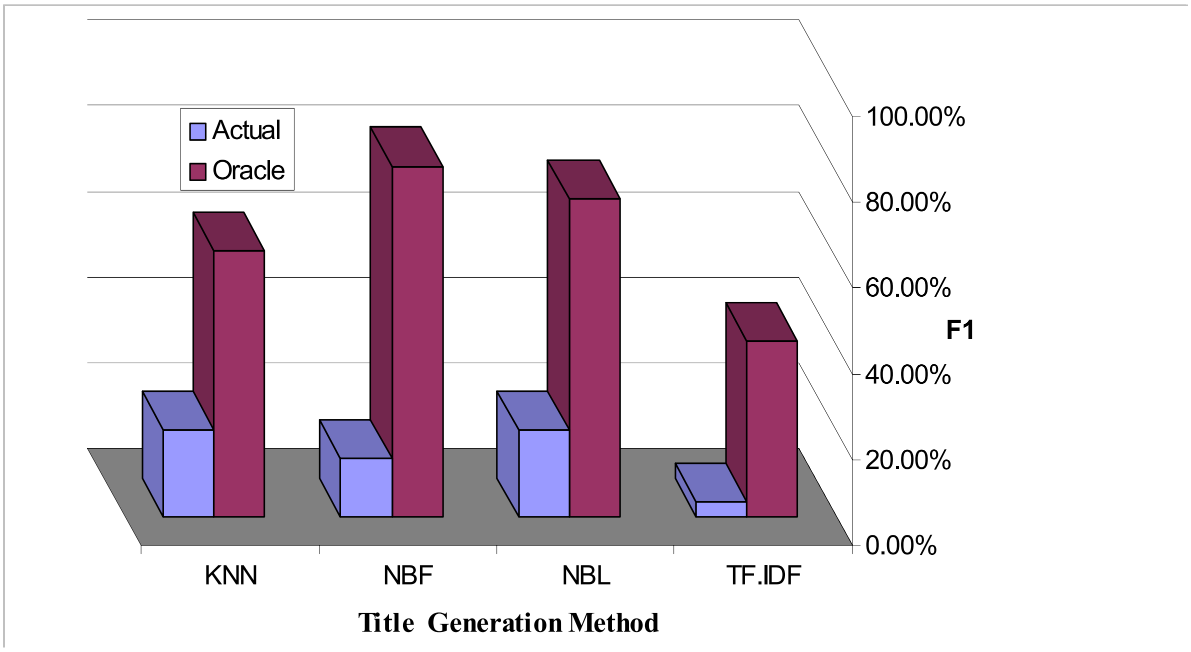
The results are shown in Figure 1. The extractive summarization approach based on TF.IDF performs worst at 3.2%, while the K-nearest neighbor (KNN) and Naïve Bayesian with limited vocabulary (NBL) approaches select the best title words at 20.04% and 20.2% respectively. The NBF approach is in the middle with 13.6%. The results of an 'oracle' approach using each method are also plotted. The oracle approach assumes that the correct title is known. In the KNN (62%) approach, the oracle will select the best title from the training corpus that matches the test title. In NBL (81.6%) and NBF (74%), the oracle selected all words from the respective vocabulary that were matches in the current title. In the TF.IDF (41%) extractive summarization approach, that sentence from the document, which most closely matches the desired title, is selected.

Sentence-based extractive summarization performs poorly. TF.IDF extractive summarization will work well if there is much redundancy in the data and the summarization does not approach less than 10% of the document size [11]. Furthermore, the extraction-based approach is unable to take full advantage of knowledge embedded in the training dataset, beyond the IDF computation. Knowledge of LT and JL is completely ignored.

KNN works surprisingly well. KNN generates titles for a new document by choosing from the titles in the training corpus. This works fairly well because the training set and test sets were constructed to guarantee good overlap in content coverage. We conclude that KNN will be good candidate for title generation if there is strong overlap in coverage between the training and test data. From the knowledge viewpoint, KNN grasps part of knowledge in LD and JL, which is much better than sentence-based extractive TF.IDF approach. However, it is a simplistic method from the generating viewpoint because it has no flexibility in generating a new title. If consideration of human readability matters, which our F1 scores did not reflect, we would expect that KNN to out-perform all the other approaches since it is guaranteed to generate human readable title.

NBF performs much worse than NBL. The difference between NBF and NBL is that NBL assumes a document word can only generate a title word with the same surface string. From the knowledge viewpoint, it should be losing information with this very strong assumption. However, the results tell us that some information can be ignored. In NBF, nothing distinguishes between important words and trivial words, and the co-occurrence between all document words and title words is measured equally. This lets frequent, but unimportant words dominate the document-word-title-word correlation. As an extreme example, stop words show up frequently in every document. However, they have little effect on choosing title words.

Thus, even though NBF seems to exploit more knowledge than NBL, it introduces more noise by not limiting the effects of frequent, but unimportant words. This conflict suggest a strategy which neither ignores the knowledge nor overemphasizes frequent, unimportant words, which we will discuss in the next section.



3 Conclusion and Future Work

From the analysis discussed in previous section, we draw the following conclusions:

1. Applying extractive summarization approaches to title generation is not good idea because extractive summarization fails to make use of training information and restricts the title candidates to the sentences of the original document. Even optimal (oracle) sentence selection provides limited performance.
2. The KNN approach works well especially when overlap in content between training dataset and test collection is large. In WWW applications, we can easily large, relevant training corpora of documents with titles, so that there is a good chance of finding an appropriate title in the training corpus for a new, unseen document. KNN is also simple in terms of implementation. Another big advantage of KNN is that it always produces human readable titles.
3. The comparison between performance of NBF and NBL shows that we need treat important words and trivial words differently to limit the noise introduced by frequent, but trivial words.

Possible improvements can be done from two different viewpoints, i.e. a learning viewpoint and a generating viewpoint:

1. Learning viewpoint

Learn from all types of knowledge. Most methods focus on how to learn the knowledge JL, which is about the correlation between document words and title words. However, other knowledge sources, i.e. LD and LT are hardly considered. NBL and NBF make no attempt to understand knowledge LD and LT. KNN only learns a small part of LD knowledge to compute the idf for document words and completely ignores the LT knowledge. One possible improvement is to take advantage of the knowledge LT and LD when considering the document-word-document-word correlation and title-word-title-word correlation in computing document-word-title-word correlation.

Learning JL knowledge correctly. Among the methods discussed in section 2, KNN learns the JL knowledge by indexing the document-title pair in training corpus while NBL and NBF encode knowledge JL by counting the document word and title word co-occurrence. No attempt is made to understand the correlation between document word sequence and title word sequence. One way to improve it is to compute the statistics on multigrams, not just unigrams. More rigorously, we can learn the knowledge JL by building up a Hidden Markov Model (HMM) for document-title pairs in the training corpus and apply the HMM model to the new document for generating new title. In both ways, the sequence information is embedded in either an HMM model or a multi-gram.

The other problem with learning JL knowledge shared by all approaches is treating all the document words equally. KNN computes the similarity between the new document and the training documents by basically measuring the overlapping between the new document and training documents. As we pointed out in section 2.4, NBF compares poorly to NBL due because it fails to treat words differently. A better weighting schema, which considers of titles as well as documents could help KNN further improve its performance. As for NBF, one improvement could be to consider a conditional probability of generating nothing (the null word) given a document word. By assigning a higher conditional probability of generating nothing to frequent, unimportant words, we can reduce the noise introduced by trivial words.

2. Generating viewpoint

In this paper, we actually implement KNN approach as the single nearest neighbor (K=1). This has no flexibility to form new titles and restricts any new title to be identical to one in the training pool. We can relax this restriction by choosing K (K>1) most similar documents in the training pool and form a new title based on the K chosen titles. This may create a more appropriate title for a new document. The trade-off would be that the title may not be human readable anymore since the new title is synthetically composed.

References

1. Ultra-Summarization: A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries. Michael Witbrock and Vibhu Mittal, Just Research. In *Proceedings of SIGIR 99*, Berkeley, CA, August 1999
2. A trainable document summarizer. J. Kupiec, J. Pedersen, and F. Chen. In *Proceedings of ACM/SIGIR'95*, pages 68-73. ACM
3. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. In *Proceedings of SIGIR 99*, Berkeley, CA, August 1999.
4. A robust practical text summarization system. T. Strzalkowski, J. Wang, and B. Wise. In *AAAI Intelligent Text Summarization Workshop*, pages 26-30, Stanford, CA, March, 1998.
5. Automatic text structuring and summary. Gernard Salton, A. Singhal, M. Mitra, and C. Buckley. *Info. Proc. And Management*, 33(2):193-207, March, 1997.
6. Automatic text summarization by paragraph extraction. M. Mitra, Amit Sighal, and Chris Buckley. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain.
7. The mathematics of statistical machine translation: Parameter estimation. Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. *Computational Linguistics*, (2):263-312, 1993.
8. An example-based mapping method for text classification and retrieval. Yang, Y., Chute, C.G. ACM Transactions on Information Systems (TOIS),12(3):252-77. 1994.
9. Information Retrieval. Chapter 7. Van Rjiesbergen. Butterworths, London, 1979. The SMART Retrival System: Experiments in Automatic Document Proceeding. Edited by Gerard Salton. Prentice Hall, Englewood Cliffs, New Jersey. 1971.
10. Selecting Text Spans for Document Summaries: Heuristics and Metrics. Vibhu Mittal, Mark Kantrowitz, Jade Goldstein and Jaime Carbonell. *AAAI-99*.
11. Nye, H. (1984). The Use of a One Stage Dynamic Programming Algorithm for Connected Word Recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. AASP-32, No 2, pp. 262-271, April 1984.
12. Kennedy, P and Hauptmann, A.G., Automatic Title Generation for the Informedia Multimedia Digital Library, ACM Digital Libraries, DL-2000, San Antonio Texas, May 2000, in press.
13. Primary Source Media, Broadcast News CDROM, Woodbridge, CT, 1995, 1996, 1997, 1998