Language Model-Based Sentence Classification for Opinion Question Answering Systems

Saeedeh Momtazi*, Dietrich Klakow†

Spoken Language Systems
Saarland University
Saarbruecken, Germany
*saeedeh.momtazi@lsv.uni-saarland.de
†dietrich.klakow@lsv.uni-saarland.de

Abstract—In this paper, we discuss an essential component for classifying opinionative and factual sentences in an opinion question answering system. We propose a language model-based approach with a Bayes classifier. This classification model is used to filter sentence retrieval outputs in order to answer opinionative questions. We used *Subjectivity dataset* for our experiments and applied different state-of-the-art smoothing methods. The results show that our proposed technique significantly outperforms current standard classification methods including support vector machines. The accuracy is improved from 90.49% to 93.35%.

I. INTRODUCTION

QUESTION answering is the task of finding natural language answers to natural language questions. Such systems have become one of the active topics in natural language processing over the past few years. Its popularity stems from the fact that a user receives an exact answer to his questions rather than being overwhelmed with a large number of retrieved documents, which he must then sort through to find the desired answer.

Among different question answering systems, a system which can answer opinion questions has been widely discussed recently, because humans like to express their opinions and are eager to know others' opinions. Motivation for this task comes from the desire to provide tools to analyze the information for individuals, governmental organizations, commercial companies, and political groups, who want to automatically track attitudes and feelings in on-line resources. What do students like about Wikipedia? How do people feel about recent events in the Middle East? Who likes Microsoft products? How do Americans consider the US-Iraq war? What organizations are against universal health care? What are the public opinions on human cloning? What users prefer Google Mail?

A system that could automatically identify opinions and emotions from text would be an enormous help to someone trying to answer these kinds of questions. Natural language processing applications could benefit from being able to distinguish between factual and opinionative information. Question answering systems which can detect and classify factual and opinionative information offers distinct advantages in deciding what information to extract and how to organize and present this information. Such system aims to present multiple answers to the user based upon opinions derived from blogs. Since most of the state-of-the-art question answering systems serve the needs of answering factual questions, opinion questions revealing answers about peoples opinions have longer and more complex answers. Therefore, they tend to scatter across different documents. Traditional question answering approaches are not effective enough to retrieve answers for opinion questions as they have been for factual questions. Hence, an opinion question answering system is essential and urgent.

To achieve such a system, a document retrieval component together with a sentence retrieval component seems a good way to provide the relevant information which can be used in the further steps of answer extraction. However, even by having a very appropriate retrieval engine there is no guarantee to retrieve opinionative sentences in top ranks. As a result, in order to be able to answer opinion questions, it is necessary to detect and classify opinionative and factual sentences retrieved by the sentence retrieval engine. An accurate classification component can offer distinct advantages in deciding what information should be retrieved and presented. This system aims to present multiple answers to the user based upon opinions derived from different sources.
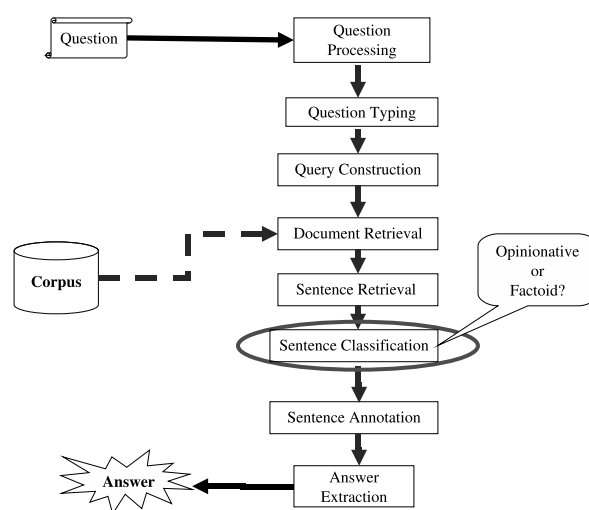

Fig. 1. The Structure of a Question Answering System

Figure 1 shows how we can benefit from a sentence classifier among a question answering system in order to distinguish between sentences that are eligible for opinionative questions and the ones are appropriate for factual questions.

In this research we expand our language model-based sentence retrieval component with a new language model-based technique to deal with opinion questions. This new step classifies all sentences into opinionative and factual, and ranks the sentences according to the classification result. A Bayes classifier with Language Models (LM) is used as the categorization paradigm in our system. The main advantage of using an LM-based approach is the large supply of known techniques to calculate and smooth probabilities, which will be discussed in this paper.

The structure of the paper is as follows: in Section 2, we describe our categorization model and how it can be used for different *n*-gram levels. Section 3 deals with the state-of-the-art smoothing techniques applied in our system and discusses the background models used in this task. In Section 4, the dataset and the baseline of our experiments are explained and then parameter study and final results are presented. Section 5 presents how our model can demonstrated as an individual component to be used in other applications. Finally, Section 6 summarizes the paper and suggests future work.

II. LANGUAGE MODEL-BASED OPINION CLASSIFICATION

Statistical LM has been successfully used in many natural language processing tasks including speech recognition [1], part of speech tagging, syntactic parsing [2], and machine translation [3]. LM-based information retrieval has received researchers' attention in the recent years. Ponte and Croft [4] showed that this method outperforms other information retrieval methods. Merkel and Klakow [5] used this technique for the sentence retrieval task; they achieved better performance than other methods. In this research we propose the same technique for the task of classifying sentences as opinionative and factual in order to deal with opinionative questions in our question answering system.

As mentioned, a Bayes classifier is used for categorization, since this classifier provides the minimum error rate if all probabilities are exactly known. Bayes classifier is defined as follows:

$$\hat{c} = \operatorname{argmax}_c P(S|c)P(c) \qquad (1)$$

where $P(c)$ is the prior probability of class c which in our case is labeled as opinionative or factual. In contrast to most of the current methods which consider $P(c)$ as a uniform distribution, we use the unigram model for this probability. Since both types of sentences, opinionative and factual, are available in the training data, we do not need to smooth this probability. $P(S|c)$ is the conditional probability of sentence S given class c. In the case of unigram language model, $P(S|c)$ is calculated as follows:

$$P^{Uni}(S|c) = \prod_{i=1...m} P(w_i|c) \qquad (2)$$

where $S = w_1...w_m$.

In order to have bigram and trigram models, $P(S|c)$ are defined as follows:

$$P^{Bi}(S|c) = P(w_1|c) \prod_{i=2...m} P(w_i|w_{i-1}c) \qquad (3)$$

$$P^{Tri}(S|c) = P(w_1|c) \; P(w_2|w_1c) \\ \prod_{i=3...m} P(w_i|w_{i-2}w_{i-1}c) \qquad (4)$$

To avoid zero probability in calculating the probabilities, we need to use smoothing methods which will be described in the next section.

III. MODELS

A. Smoothing Models

Three different smoothing methods were presented by Zhai and Lafferty [6] for the information retrieval task. In the following sections, we describe these standard methods for the task of sentence classification.

1) Jelinek-Mercer: Jelinek and Mercer [7] introduced a linear interpolation technique.

$$P_\lambda(w_i|c) = \lambda \frac{N(w_i,c)}{\Sigma_{w_i}N(w_i,c)} + (1-\lambda)P_{BG}(w_i) \qquad (5)$$

where λ is the smoothing parameter to be determined. $N(w_i,c)$ is the count of word w_i in class c and $P_{BG}(w_i)$ is the background probability which will be described in Section III-B.

2) Bayesian Smoothing with Dirichlet Prior: If the LM is considered as a multinomial distribution and the Dirichlet distribution is used as the conjugate prior, then the model given in [8] is:

$$P_\mu(w_i|c) = \frac{N(w_i,c) + \mu P_{BG}(w_i)}{\Sigma_{w_i}N(w_i,c) + \mu} \qquad (6)$$

where μ is the smoothing parameter to be tuned on the development data. As in the Jelinek-Mercer method, $N(w_i,c)$ is the count of word w_i in class c and $P_{BG}(w_i)$ is the background probability.

3) Absolute Discounting: In the absolute discounting method a very small constant is subtracted from the probability of seen events and is distributed over all seen and unseen events [9].

$$P_\delta(w_i|c) = \frac{\max(N(w_i,c) - \delta, 0)}{\Sigma_{w_i}N(w_i,c)} \\ + \frac{\delta B}{\Sigma_{w_i}N(w_i,c)}P_{BG}(w_i) \qquad (7)$$

where $N(w_i,c)$ and $P_{BG}(w_i)$ are calculated as in previous methods. δ is the smoothing parameter and B denotes how often $N(w_i,c)$ is larger than δ.

B. Background Models

To use the above smoothing methods, we need background models which will be described in this section.

1) Zerogram: The simplest model is the zerogram model which is the uniform distribution of words. The zerogram model can be calculated as follows:

$$P_{BG}^{Zero}(w_i) = \frac{1}{|V|} \qquad (8)$$

where $|V|$ is the vocabulary size.

We used this model as the background of our unigram model.

2) Unigram: Another background model used in our research is the unigram model. The unigram model is computed with maximum likelihood estimation and is used as the background of our bigram model.

$$P_{BG}^{Uni}(w_i) = \frac{N(w_i)}{\Sigma_{w_i} N(w_i)} \qquad (9)$$

where $N(w_i)$ is the frequency of the word w_i in the corpus.

3) Bigram: The bigram model is also used as one of the background models in this paper. Our trigram model is smoothed with the bigram model calculated as follows:

$$P_{BG}^{Bi}(w_i) = \frac{N(w_{i-1}, w_i)}{N(w_{i-1})} \qquad (10)$$

where $N(w_{i-1}, w_i)$ is the frequency of the sequence $w_{i-1} w_i$ and $N(w_{i-1})$ is the frequency of the word w_{i-1} in the corpus.

IV. EXPERIMENTS

A. Dataset

For our experiments, we used an automatically labeled dataset, named *Subjectivity datasets*, created at Computer Science Department of Cornell University and released on June 2004 [10]. This dataset[1] contains 10000 sentences in movie domain. 5000 of the sentences were snippets of movie reviews from Rotten Tomatoes (*http://www.rottentomatoes.com/*) which considered as subjective (opinionative) sentences and 5000 of them are plot summaries for movies from the Internet Movie Database (*http://www.imdb.com*) which counted as objective (factual) sentences. The dataset has been selected only from sentences or snippets at least ten words long and drawn from reviews or plot summaries of movies released post-2001.

We divided the dataset into three non-overlapping parts. 80% of this dataset was used as training data; 10% as development set; and 10% for testing. The smoothing parameters were tuned on the development set and the results were achieved by applying the tuned parameters on the test set. For the evaluation, we did 10 fold cross-validation on our dataset. As an evaluation metric for our experiments, we calculate the accuracy of classification which is as follows:

$$Accuracy = \frac{\text{number of sentences classified truly}}{\text{number of all sentences}} \qquad (11)$$

[1]available at http://www.cs.cornell.edu/people/pabo/movie-review-data/

B. Baseline

In order to compare our results with other state-of-the-art methods, we chose the Support Vector Machine (SVM) classifier as our baseline, since this method is the best traditional method in literature [11], [12]. We used the SVM-Light [13] implementation for our baseline experiments. Four different features have been used in this experiment. Two of them used the word unigram in which the first one considers the presence of words instead of their frequencies and the second one uses the frequency of words. We repeated these two experiment by adding bigram features to unigram. The best accuracy of the SVM in our data set for each of the features are presented in Table 1.

TABLE I
THE ACCURACY OF SVM WITH DIFFERENT FEATURES

	Unigram	Unigram+Bigram
Word Presence	90.11	**90.49**
Word Frequency	89.92	90.47

The table shows that in both unigram and unigram+bigram models the presence of words results in better accuracy than the frequency of words as it has been reported by Pang [11]. The maximum accuracy of SVM is 90.49% which is achieved by applying the combination of unigram and bigram models with considering the word presence.

C. Study of Parameters

The results of our experiments on the development set are presented in this section. In this part we used three different smoothing methods described in Section III-A and applied them on unigram, bigram and trigram language models over different smoothing parameters to find the best values of each parameter.

1) Jelinek-Mercer: The experiments with Jelinek-Mercer smoothing over different interpolation weights is presented in Figure 2.
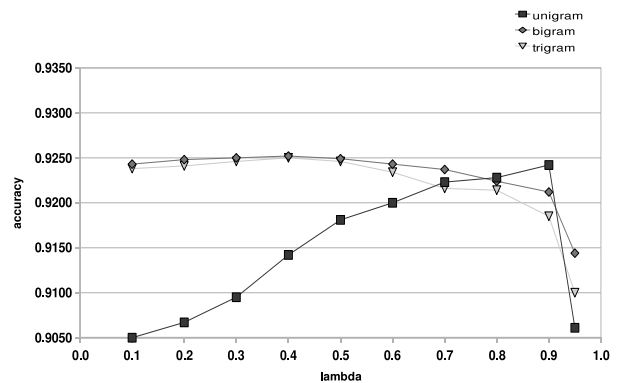

Fig. 2. Accuracy of classification for different Jelinek-Mercer smoothing parameters

The figure shows the results of Jelinek-Mercer method by using different *n*-gram models including unigram, bigram and

TABLE II
COMPARISON THE ACCURACY OF DIFFERENT LM-BASED APPROACHES FOR OPINION SENTENCE CLASSIFICATION.

(In contrast, the best SVM has 90.49% accuracy)

Smoothing	Unigram		Bigram		Trigram	
	Accuracy	%change	Accuracy	%change	Accuracy	%change
Jelinek-Mercer	92.51	+2.23	92.57	+2.30	92.56	+2.29
Dirichlet Prior	92.59	+2.32	*93.35*	+3.16	93.01	+2.78
Absolute Discounting	92.45	+2.17	93.15	+2.94	92.90	+2.66

trigram. We can see that the accuracy of unigram model completely changes over different values of λ and the best accuracy achieved by $\lambda = 0.9$. The accuracy of bigram is almost the same as the accuracy of trigram model and both of them reach their maximum accuracy around $\lambda = 0.4$.

2) Bayesian Smoothing with Dirichlet Prior: Figure 3 shows the accuracy of sentence classification with Dirichlet Prior over different values of the smoothing parameter.
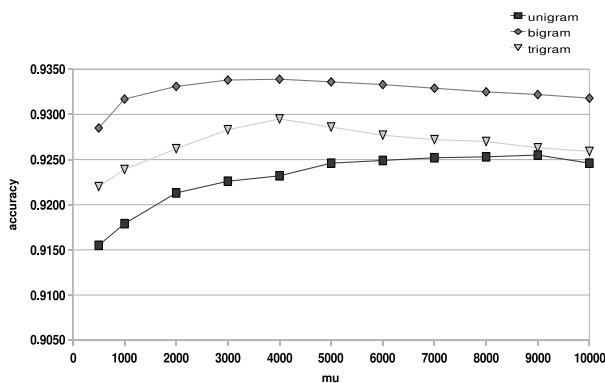

Fig. 3. Accuracy of classification for different Dirichlet Prior smoothing parameters

The accuracy curve of the unigram model in this figure is smoother than the Jelinek-Mercer method. As shown in the figure, by increasing the smoothing parameter μ, the accuracy of unigram model increases and the maximum value is received when $\mu = 9000$. However, bigram and trigram model perform better with the smaller values of μ and the bigram model significantly outperforms other n-gram models.

3) Absolute Discounting: Absolute discounting is another smoothing method which we use for our experiments and present its results in Figure 4.

In this smoothing method, like the former methods, the unigram model has a lower accuracy than bigram and trigram models. The behavior of bigram and trigram are also the same as other smoothing methods, since the maximum accuracy of the bigram and trigram models achieved by the same value of the smoothing parameter $\delta = 0.9$, while the unigram performs best with $\delta = 0.3$.

D. Results

After using the development data for all of the smoothing method, the tuned parameters were applied on the test set. Table 2 shows the results of our experiments. According to
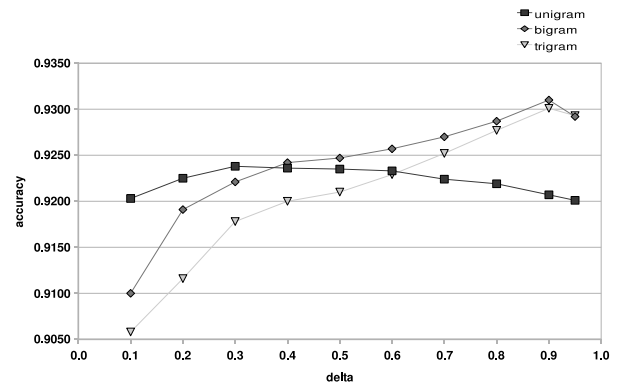

Fig. 4. Accuracy of classification for different Absolute Discounting parameters

this table, the bigram model performs better than the unigram model which proves the bigram information is more useful than the unigram for sentence classification. The bigram model also performs better than the trigram model, which indicates that the trigram model is too sparse on sentence level and it can not work properly for the sentence classification task. Among different smoothing methods, Dirichlet prior with a bigram model performs the best and achieves 93.35% accuracy in opinion sentence classification.

Comparing our results to the baseline presented in Table 1, it is obvious that all described LM-based models outperform the best result we achieved by SVM. In Table 2, the columns labeled as "%change" show the difference between the accuracy of our models and the best accuracy of SVM. All differences are statistically significant according to *t*-test at the level of *p-value* < 0.01.

V. SYSTEM DEMONSTRATION

As mentioned before, the proposed model is implemented as a component in a question answering system, in which the output of sentence retrieval can be used as an input for our sentence classifier. This new module should classify the sentences either as factual or as opinionative.

However, this model can also be used individually for any other applications which require an accurate classification at sentence level based on their subjectivity. In order to show how our language model-based sentence classifier works, a simple demonstration of this system has been provided. First, system asks the user either to enter a single sentence or a set of sentences for testing the model. If the user wants to test

the model with a single sentence, a text box will be pop up in which the user can type a test sentence; otherwise, there is an option that the user can browse a file as an input. Then, the test sentence(s) should be labeled as opinionative or factual. Finally, if the input is a single sentence, the results of the classifier will appear in the screen; otherwise, the results will be written in a file. In the output, six results are reported based on three different smoothing techniques (Jelinek-Mercer, Dirichlet Prior, and Absolute Discounting) and two n-gram levels (unigram and bigram). Each result represents whether the coresponding model has classified the input sentence(s) as opinionative or factual while presenting the confidence score.

VI. CONCLUSION

In this paper we presented a language model-based approach for classifying sentences as opinionative and factual in the context of opinion question answering. We used a Bayes classifier with different smoothing methods and different n-gram models. The results show that our proposed approach significantly improves the sentence classification performance and outperforms the SVM which is the best categorization method in the available literature [11], [12].

In this research we used all of the words of sentences as sentences' features. In future work, we plan to apply different feature selection techniques and evaluate their effects on the sentence classification performance.

REFERENCES

[1] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge MA: The MIT Press, 1998.
[2] E. Charniak, *Statistical Language Learning*. Cambridge MA: The MIT Press, 1993.
[3] P. Brown, J. Cocke, S. Pietra, V. D. Pietra, F. Jelinek, J. Laerty, R. Mercer, and P. Roossin, "A statistical approach to machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, 1990.
[4] J. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *ACM SIGIR Conference Proceedings*, 1998, pp. 275–281.
[5] A. Merkel and D. Klakow, "Comparing improved language models for sentence retrieval in question answering," in *Computational Linguistics in the Netherlands Conference Proceedings*, 2007.
[6] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *ACM SIGIR Conference Proceedings*, 2001.
[7] F. Jelinek and R. Mercer, "Interpolated estimation of markov source parameters from sparse data," in *Proceedings of an International Workshop on Pattern Recognition in Practice*, 1989.
[8] D. Mackay and B. Peto, "A hierarchical dirichlet language model," *Natural Language Engineering*, vol. 1, no. 3, pp. 1–19, 1995.
[9] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependencies in language modelling," *Computer Speech and Language*, vol. 8, pp. 1–38, 1994.
[10] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *ACL Conference Proceedings*, 2004.
[11] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *EMNLP Conference Proceedings*, 2002.
[12] W. Zhang, C. Yu, and W. Meng, "Opinion retrieval from blogs," in *ACM CIKM Conference Proceedings*, 2007, pp. 831–840.
[13] T. Joachims, *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*. MIT-Press, 1999.