Opinion Retrieval Experiments using Generative Models: Experiments for the TREC 2007 Blog Track

Yuki Arai[1] and Koji Eguchi[1,2]
[1] Kobe University, Kobe, Japan
[2] National Institute of Informatics, Tokyo, Japan
yuuki20@cs25.scitec.kobe-u.ac.jp
eguchi@port.kobe-u.ac.jp

Abstract

Ranking blog posts that express opinions regarding a given topic should serve a critical function in helping users. We explored a couple of methods for opinion retrieval in the framework of probabilistic language models. The first method combines topic-relevance model and opinion-relevance model, at *document* level, that captures topic dependence of the opinion expressions. The second method combines the aforementioned topic-opinion relevance models at *sentence* level, and accumulates the negative cross entropy between the combined relevance models and each sentence model to obtain a document-level score. This paper reports the overview of our methods and the evaluation results on the Opinion Retrieval Task at the TREC 2007 Blog Track.

1 Introduction

The recent rapid expansion of access to information has significantly increased the demands on retrieval or classification of sentiment information from a large amount of textual data. The field of *sentiment classification* has recently received considerable attention, where the polarities of sentiment, such as positive or negative, were identified from unstructured text [8]. A number of studies have investigated sentiment classification at document level, e.g., [7, 2], and at sentence level, e.g., [4, 5, 6]; however, the accuracy is still less than desirable. Therefore, ranking according to the likelihood of containing sentiment information is expected to serve a crucial function in helping users.

For this objective, Eguchi and Lavrenko [3] proposed *sentiment retrieval* models, aiming at finding information with a specific sentiment polarity on a certain topic, where the topic dependence of the sentiment was considered. Intuitively, the expression of sentiment in text is dependent on the topic. Sentiment polarities are also dependent on topics or domains. A couple of examples follow. A negative view for some voting event may be expressed using 'flaw', while a negative view for some politician may be expressed using 'reckless'. As another example, the adjective 'unpredictable' may have a negative orientation in an automotive review, in a phrase such as 'unpredictable steering', but it could have a positive orientation in a movie review, in a phrase such as 'unpredictable plot', as mentioned in [9] in the context of his sentiment word detection. Eguchi and Lavrenko's sentiment retrieval models can address both cases based on the framework of generative language modeling, not only assuming query terms expressing a certain topic, but also assuming that the sentiment polarity of interest is specified by the user in some manner.

For the TREC 2007 Blog Track, we followed [3], but we set aside the topic dependence of the sentiment *polarities* and focused on that of the sentiment *expressions*. In [3], sentence level was focused in the experiments; however, the model can be applied to textual chunks of any length. Therefore, we combines topic-relevance model and opinion-relevance model, at *document* level, that captures topic dependence of the opinion expressions. We also combines the aforementioned topic-opinion relevance models at *sentence* level, and accumulates the negative cross entropy between the combined relevance models and each sentence model to obtain a document-level score.

2 A Generative Model of Opinion

2.1 Definitions

According to [3], we start by providing a set of definitions that will be used in the remainder of this section. The task of our model is to *generate* a collection of statements $\mathbf{w}_1 \ldots \mathbf{w}_n$. A statement $\mathbf{w}_i$ is a string of words $\mathbf{w}_{i1} \ldots \mathbf{w}_{in_i}$, drawn from a common vocabulary $\mathcal{V}$. We introduce a binary variable $b_{ij} \in \{S, T\}$ as an indicator of whether the word in the jth position of the ith statement will be a topic word or an opinion-bearing word. For our purposes, b_{ij} is determined heuristically (*automatic annotation*), in this paper.

As a matter of convenience we will often denote a statement as a pair $\{\mathbf{w}_i^s, \mathbf{w}_i^t\}$, where $\mathbf{w}_i^s$ contains the opinion-bearing words and $\mathbf{w}_i^t$ contains the topic words. As we mentioned above, the user's query is treated as just another statement. It will be denoted as a pair $\{\mathbf{q}^s, \mathbf{q}^t\}$, corresponding to opinion-bearing words and topic keywords. We will use $\mathbf{p}$ to denote a unigram language model, i.e., a function that assigns a number $\mathbf{p}(v) \in [0, 1]$ to every word v in our vocabulary $\mathcal{V}$, such that $\Sigma_v \mathbf{p}(v) = 1$. The set of all possible unigram language models is the probability simplex $\mathbb{P}$. We define $\pi : \mathbb{P} \times \mathbb{P} \to [0, 1]$ to be a measure function that assigns a probability $\pi(\mathbf{p}_1, \mathbf{p}_2)$ to a pair of language models $\mathbf{p}_1$ and $\mathbf{p}_2$.

2.2 Generative model

Using the definitions presented above, and assuming that $\pi()$ is given, we hypothesize that a new statement $\mathbf{w}_i$ containing words $\mathbf{w}_{i1} \ldots \mathbf{w}_{im}$ can be generated according to the following mechanism.

1. Draw $\mathbf{p}_t$ and $\mathbf{p}_s$ from $\pi(\cdot, \cdot)$.
2. For each position $j = 1 \ldots m$:
   (a) if $b_{ij} = T$: draw $\mathbf{w}_{ij}$ from $\mathbf{p}_t(\cdot)$;
   (b) if $b_{ij} = S$: draw $\mathbf{w}_{ij}$ from $\mathbf{p}_s(\cdot)$.

The probability of observing the new statement $\mathbf{w}_{i1} \ldots \mathbf{w}_{im}$ under this mechanism is given by:

$$\sum_{\mathbf{p}_t, \mathbf{p}_s} \pi(\mathbf{p}_t, \mathbf{p}_s) \prod_{j=1}^{m} \begin{cases} \mathbf{p}_t(\mathbf{w}_{ij}) & \text{if } b_{ij} = T \\ \mathbf{p}_s(\mathbf{w}_{ij}) & \text{otherwise} \end{cases} \qquad (1)$$

We use this simple equation instead of that in [3] since we can set aside sentiment polarities in this paper. The summation in equation (1) goes over all possible pairs of language models $\mathbf{p}_t, \mathbf{p}_s$, but we can avoid integration by specifying a mass function $\pi()$ that assigns nonzero probabilities to a finite subset of points in $\mathbb{P} \times \mathbb{P}$. We accomplish this by using a nonparametric estimate for $\pi()$, the details of which are provided below.

2.3 Using the model for retrieval

The generative model presented above can be applied to opinion retrieval in the following fashion. Following [3], we start with a collection of statements C and a query $\{\mathbf{q}^s, \mathbf{q}^t\}$ supplied by the user, where $\mathbf{q}^s$ can be some typical opinion-bearing words with either positive or negative polarity and $\mathbf{q}^s$ can be words in the title field in the topic given by the Blog Track organizers. We use the procedure outlined in Section **2.2** to estimate the topic- and opinion-relevance models corresponding to the user's information need, and then determine which statements in our collection most closely correspond to these models of relevance. The topic-relevance model R_t and opinion-relevance model R_s are estimated in the similar fashion described in [3] for each query $\{\mathbf{q}^s, \mathbf{q}^t\}$. Once we have estimates for the topic and sentiment relevance models, we can rank testing statements $\mathbf{w}$ by their similarity to R_t and R_s. We rank statements using a variation of cross-entropy, which was proposed by [10] and modified for sentiment retrieval task in [3]:

$$\alpha \sum_v R_t(v) \log \mathbf{p}_t(v) + (1-\alpha) \sum_v R_s(v) \log \mathbf{p}_s(v). \tag{2}$$

Here the summations extend over all words v in the vocabulary. A weighting parameter α allows us to change the balance of topic and sentiment in the final ranking formula; its value can be selected empirically.

3 Opinion Retrieval Task

3.1 Method-1: Using topic-opinion relevance models at document level

We define a variation of the sentiment retrieval model [3]. As input, we used (1) a set of topic keywords $\mathbf{q}^t$ and (2) a set of opinion-bearing seed words $\mathbf{q}^s$.

We detected opinion-bearing words from each document using lists of words. We used sentiment word list contained in *OpinionFinder* [1], which consists of 2230 positive and 3913 negative words. We extracted opinion-bearing expressions using the list of words above to construct opinion-relevance models.

3.2 Method-2: Using topic-opinion relevance models at sentence level

We also constructed topic-relevance model and opinion-relevance model at sentence level, not at document level that was discussed in Section **3.1**. Following Section 3.1, we used the same queries $\mathbf{q}^t$ and $\mathbf{q}^s$, and the same way of detecting opinion-bearing words from target text. We accumulated the (negative) cross-entropy between the relevance models and each sentence model to obtain a document-level score, by summing up the sentence-level cross-entropy over the whole document.

4 Results and Discussions

According to the evaluation results, Method-1 worked but Method-2 did not work well. Detailed analysis on the evaluation results is ongoing. Using the relevance judgment data given by the organizers, we are currently investigating to estimate the model parameters appropriately for the task defined in the TREC Blog Track, and to perform the additional experiments.

Acknowledgments

This work was supported in part by the Grant-in-Aid for Scientific Research (#17680011, #18650057 and #04560004) from the Ministry of Education, Culture, Sports, Science and Technology, Japan. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the sponsor.

References

[1] MPQA releases – corpus and opinion recogntion system. http://www.cs.pitt.edu/mpqa/.

[2] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of WWW-03, 12th International Conference on the World Wide Web*, pages 519–528, Budapest, HU, 2003. ACM Press.

[3] K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Jul. 2006.

[4] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of KDD '04, the ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, Seattle, US, 2004. ACM Press.

[5] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings COLING-04, the Conference on Computational Linguistics*, Geneva, CH, 2004.

[6] K. Nigam and M. Hurst. *Computing Attitude and Affect in Text: Theory and Applications*, chapter Towards a Robust Metric of Opinion. Springer, 2005.

[7] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pages 79–86, Philadelphia, Pennsylvania, USA, Jul. 2002.

[8] J. Shanahan, Y. Qu, and J. Wiebe, editors. *Computing attitude and affect in text*. Springer, 2005.

[9] P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 417–424, Philadelphia, Pennsylvania, USA, Jul. 2002.

[10] C. Zhai. *Risk Minimization and Language Modeling in Text Retrieval*. PhD dissertation, Carnegie Mellon University, Pittsburgh, PA, July 2002.