Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 27 (2011) 267 – 273

Procedia Social and Behavioral Sciences

Pacific Association for Computational Linguistics (PACLING 2011)

Recommending the Meanings of Newly Coined Words

Jong Gun Lee[a*] , Young-Min Kim[b] , Jungyeul Park[c] , Jeong-Won Cha[d]

[a]France Telecom - Orange Labs, France
[b]University of Avignon, France
[c]ELDA/ELRA, France
[d]Changwon National University, Republic of Korea

Abstract

In this paper, we investigate how to recommend the meanings of newly coined words, such as newly coined named entities and Internet jargon. Our approach automatically chooses a document explaining a given newly coined word among candidate documents from multiple web references using Probabilistic Latent Semantic Analysis [1]. Briefly, it involves finding the topic of a document containing the newly coined word and computing the conditional probability of the topic given each candidate document. We validate our methodology with two real datasets from MySpace forums and Twitter by referencing three web services, Google, Urbandictionary, and Wikipedia, and we show that we properly recommend the meanings of a set of given newly coined words with 69.5% and 80.5% accuracies based on our three recommendations, respectively. Moreover, we compare our approach against three baselines where one references the result from each web service and our approach outperforms them.

Keywords: Newly Coined Word, Topic Model, Probabilistic Latent Semantic Analysis

1. Introduction

We live in the age of information deluge. While people can access information through these web services, at the very same time they used to face unknown words, which they do not know their meaning. A part of these kind words could be newly coined words, such as newly minted named entities and Internet jargon. When people face an unknown word, they might behave differently. One might pass by it expecting to understand its meaning with the next bit of text that comes along or another would try to find its meaning with existing resources such as dictionaries and thesauri. If a word is very newly coined, people might not be able to find its meaning within a few top ranked results from a search engine, which usually produces most frequently queried words on the top. When an online encyclopedia gives multiple definitions for an entered unknown word, one may understand its proper meaning after reading through the definitions. If an online service provides the proper explanation of an unknown word in a context, it will be extremely expedient.

* Correspondng author.
Email address: jonggun.lee@orange-ftgroup.com (Jong Gun Leea), young-min.kim@univ-avignon.fr (Young-Min Kimb), park@elda.org (Jungyeul Parkc), jcha@changwon.ac.kr (Jeong-Won Chad)

1877-7058 © 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
Selection and/or peer-review under responsibility of PACLING Organizing Committee.
doi:10.1016/j.sbspro.2011.10.607

In this paper, we address how to find and recommend the meaning of a newly coined word. A newly coined word (NCW), however, is a loosely defined term and there are many possible ways to define it. In any definition, the purpose of this paper is not how to identify or verify NCWs with given documents, but how to provide the meanings of given unknown words which meanings users are not familiar with as the words were recently minted. Additionally we limit the scope of NCWs into newly coined named entities and Internet jargon, not self-describing words such as emoticons and web addresses. Our approach to find the meaning of a NCW is motivated by an intuition - if one wants to find the meaning of a newly coined word, she has higher chance of finding it when using multiple references rather than using a specific one. So we use, given a document containing a NCW, a bag of candidate web documents from multiple resources: search engine, user-contributed dictionary, and user-contributed encyclopedia. Among the candidates, we select a relevant document using Probabilistic Latent Semantic Analysis (PLSA) method [1], which is one of the widely used probabilistic topic models. The overall process involves a) building a topic model for a given dataset, b) inferring the topic distribution of a document containing a NCW and choosing the most probable topic of the document, c) inferring the chosen topic's probability for a set of candidate documents and computing the joint probabilities between the topic and candidates, and d) recommending a relevant candidate with a maximum joint probability.

A "newly coined word" is a loosely defined term and there are many possible ways to define it. One could define it as words newly coined into a web corpus within a recent month and another could apply an occurrence threshold to words minted during the past two months. In this paper, we empirically define NCWs as follows. We, first, refer OOV words as the ones not listed in a dictionary. Then intuitively NCWs are OOV words, but the opposite is not true. It is still not trivial to select out NCWs among OOV words based on whether an OOV word is recently coined or not. Thus we assume that NCWs are not frequently occurred in the corpus among OOV words if they are recently coined. Based on this assumption, we adjust an occurrence threshold to select infrequent OOV words as NCWs and then we manually filter out infrequently occurred but globally accepted words. Consequently, we refer remained OOV words as NCWs.

2. Overall scheme

This work is rooted in two simple questions: Can we help online users' readability by providing the meanings of NCWs? Then how can we automatically find their meanings? This paper aims to find and recommand automatically the meaning for a given NCW. The types of NCWs are including, but not limited to, misspelled words, self-describing words (e.g. emoticons or web addresses), named entities, internet jargons, etc. We comfortably resolve the words in the first and second types.

In order to find the meaning of a given NCW, we collect candidate documents from the result of multiple web references, such as Internet search engines, user-contributed dictionaries, and encyclopedias, and put them into a bag. Since we do not know the current evolution stage of the NCW, i.e., we do not know where the meaning of the NCW exists, we consider the result from multiple references, which possibly contain the meaning of the NCW. In the following of this paper, we use three web services, Google, Urbandictionary, and Wikipedia, as references, but we could extend our approach by simply adding other candidate documents from additional web references into the bag.

Among candidate documents, to choose a relevant document, which is semantically close to a given document, is based on to measure the semantical similarity between the given document and a candidate. A na¨ıve way to compare the similarity between two documents is to represent them with two vectors based on word occurrences and compare the distance between the vectors; the less the distance between two documents is, the closer they are semantically. However, when the length of documents is short, it is not easy to measure the similarity between two documents by the vector-based representation way because a vector easily becomes sparse.

Thus in order to find a relevant document for a given NCW among candidate documents, we use a topic model, which extracts hidden latent topics for a set of documents. Among topic models, we use

Probabilistic Latent Semantic Analysis (PLSA) approach [1]. PLSA is a well-known probabilistic model for document clustering and it finds latent semantic topics where each topic is constituted by a set of conditional probabilities of words. So, given a dataset, a PLSA model gives us a set of hidden semantic topics, each of which is characterized by the most probable words. This model is also used for Information Retrieval (IR) tasks as it replaces a bag-of-words space of documents with a latent topic space. A PLSA model, as a generative model, assumes that each word in a document is generated by choosing an latent variable, called a semantic topic, and the generation process of a word in a document is modeled by the following joint probability of word and document:

$$p(d,w) = p(d)\sum_{z} p(z \mid d)p(w \mid z), \quad (1)$$

where d, w, and z are a document, a word, and a topic, respectively. We obtain the model parameters, p(d), p(z|d) and p(w|z) for $\forall d$, w; z, by maximizing the log-likelihood function (Eq. 2) and we use an Expectation-Maximization (EM) algorithm to maximize the equation.

$$\mathcal{L} = \sum_{d}\sum_{w} n(d,w)\log p(d,w) \quad (2)$$

With the set of parameters from the learning step, we infer the topic of an unseen document by iterating the EM steps with the fixed conditional probabilities of words, p(w|z). Since p(w|z) is already fixed in an inferring step, the time for inferring the topic of a new document is much shorter than the time for learning.

3. Description of our approach

3.1. Preprocessing and Building a topic model

Before applying a PLSA model to a document, we first preprocess the document based on the following rules. (1) Remove self-describing words Self-describing words, such as emoticons and URLs, are not related to the topic of the document. Thus, we exclude them with a set of predefined regular expressions. (2) Use only vocabulary words in learning and inferring phrases Naturally, newly coined words are continuously introduced and evolved. It means that online users keep to make new words and a NCW cannot consistently represent a certain topic. Thus, to build a robust topic model, we use only vocabulary words in learning and inferring phrases. Additionally based on this rule, we avoid introducing misspelled words into a topic model. (3) Extract a set of surrounding words near a given NCW in the inferring phrases Web documents, such as blogs, discussion threads, and chatting-style messages, are freely written by online users and mostly unstructured and unorganized. Thus, to overcome the issue coming from users' informal writing style, we use a set of surrounding words near a given NCW. This rule is also applied to find the topic of a candidate document We build a PLSA topic model with a learning dataset as described in the previous section. For each document, we apply the first and second rules mentioned in Section 3.1. Then, by maximizing Eq. (2) with EM, we find the model parameters, p(d), p(z|d) and p(w|z).

3.2. Finding the meaning of a given NCW

The estimation procedure to find the meaning of a given NCW consists of three steps: to infer the topic of the document including the given NCW, to collect a set of candidate documents which probably contain the meaning of the NCW, and to choose the most relevant document amongst candidates. As we mentioned in Section 3.1, we use a set of surrounding words near the given NCW when inferring the topic of a document. Henceforth, we explain the detailed inference procedure with the terminologies in Table 1.

Given w* and d*, we extract n vocabulary words surrounding w* and prepared d'^{*}. Then we infer the posterior probability $p(z|\, d'^{*}) \forall z \in Z$ and prior probability $p(d'^{*})$ by applying an EM algorithm with the fixed parameter p(w|z). We choose the z which $p(z|d\ d'^{*})$ is the largest as the most probable topic z_{best} of d* for the given w . We collect a set of candidate documents C (w*) for w from multiple web references. For each $c_l(w^*)$, we extract n vocabulary words surrounding w and prepare a new $c'_l(w^*)$. We select the most relevant candidate document by inferring the topics of candidate documents.

Terminology	Description
d	a learning document
w	a word
z	a latent topic
$\mathcal{D}$	a set of learning documents
$\mathcal{V}$	a set of words from $\mathcal{D}$
Z	a set of latent topics
d^*	a new document
d'^*	a set of vocabulary words surrounding a given NCW
w^*	a NCW in d^*
$c_l(w^*) \in C(w^*)$	l^{th} candidate document in a set of candidate documents $C(w^*)$ for w^*

Table 1. Terminologies

Dataset	Service	# Documents	Description
D-my-ALL	MySpace	14.7k+	Overall dataset
D-my-LRN		9.8k+	Learning dataset
D-my-TST		4.9k+	Test dataset
D-tw-ALL	Twitter	27k+	Overall dataset
D-tw-LRN		18k+	Learning dataset
D-tw-TST		9k+	Test dataset

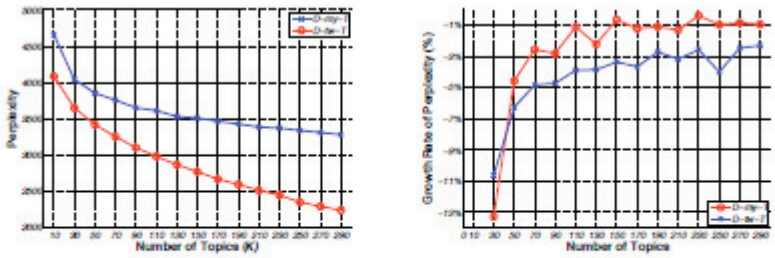
Table 2. Statistics of MySpace and Twitter datasets

For each $c'_l(w^*)$, we infer the probability $p(z|c'_l(w^*)) \quad \forall z \in Z$ and $p(c'_l(w))$ and then pick . For all $p(z_{best}|c'_l(w^*))$ candidates, we compute $p(z_{best}|c'_l(w))\ p(c'_l(w))$, which means the joint probability of z_{best} and $c'_l(w)$. Among all these values, we choose the candidate document which has the largest joint probability.

4. Data set

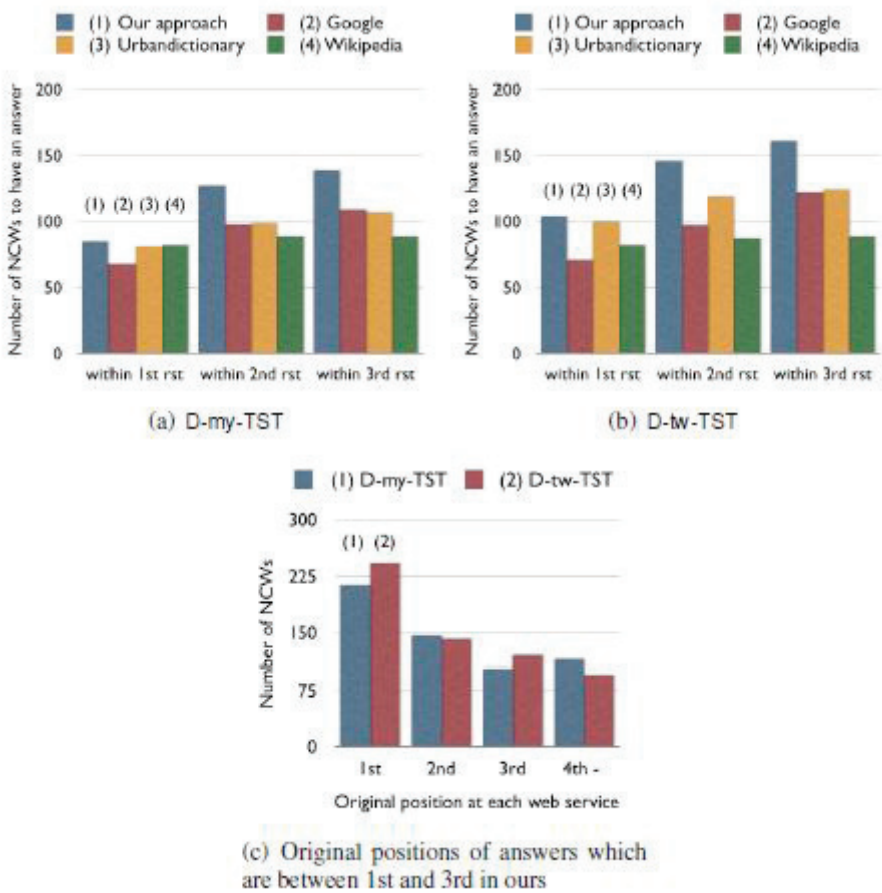
For the experiment, we use two datasets crawled from MySpace discussion forums and Twitter[1] MySpace is one of the representative online social networking services. Its online users can make friends, share their contents, such as photos and videos, with friends, and communicate with others. Twitter is an emerging online service to exchange short chatting messages with friends, namely called followers and followees, and it supports mobile interfaces as well as web interfaces. In the rest of this paper, we call MySpace and Twitter datasets as D-my-ALL and D-tw-ALL respectively and in Table 2 we present the brief statistics on the numbers of documents. D-my-ALL was crawled from MySpace forums and it contains about 16,000 discussion threads. Among these threads, we exclude threads with only few words and we use about 14,700 threads for our experiment. D-tw-ALL is composed of about 27,000 Twitter messages. For each dataset, we use randomly selected 2/3 as a learning dataset and the rest as a test dataset.

[1] Fundaci´on Barcelona Media (FBM) crawled and distributed datasets for Content Analysis for WEB 2.0 (CAW 2.0) workshop.http://caw2.barcelonamedia.org/



(a) Perplexity (b) Growth Rate of Perplexity

Fig. 1. Result of perplexity (*D-my-T*: D-my-TST and *D-tw-T*: D-tw-TST)



(c) Original positions of answers which are between 1st and 3rd in ours

Fig. 2. (a) and (b) - the number of NCWs to exist an answer within ith result (rst.) from the given system, (c) - the original positions of answers positioned between 1st and 3rd recommendations in our system

4.1. Experiment results with our approach

Building a topic model: Before building a topic model, we first should decide the number of topics for each learning dataset because PLSA approach builds a topic model for a given number of topics. Among the state-of-the-art to deal with this open question, which is to determine the number of topics in topic mining, we adopt a perplexity measure [1], which is used for measuring the performance of a PLSA model by evaluating the probabilistic parameters from a learning model against a test dataset. By the definition of perplexity, its value is decreasing as the number of topics is increasing as shown in Figure 1(a). When the number of topics is small, the decreasing rate is much but as the number is increasing its decreasing rate is getting less. For a close look for the decreasing ratio, we plot the growth rate of perplexity values in Figure 1(b). Based on this figure, we decide the numbers of topics of D-my-LRN and D-tw-LRN as 50 and 70 where each growth rates is less than -5%, respectively. We stop the iteration of EM update process when the increment of log-likelihood reaches our convergence condition, e^{-5}. With these parameters, the number of topics and the convergence condition of EM iteration, we build topic models for D-my-LRN and D-tw-LRN.

Finding the meaning of a given NCW: It is well-known that online users pay more attention and click more often top-ranked result. Agichtein et al. [2] aggregated statistics of user search sessions and presented the relative click frequencies for top-30 result positions. When the click frequency of top-1 result is 1, the relative frequency of top-2 (top-3) result is 0.55 0.6 (0.45 0.5, respectively). Moreover, they mentioned that the relative frequency of top-4 result is only about 0.3. In [3], Joachims and Radlinski reported that "for all results below the third rank, users didn't even look at the result for more than half of the queries". Thus, in the following experiment we measure the performance of our proposed methodology with top-3 recommendation result. For our experiment, we select 200 NCWs from each test dataset by extracting me-dially ranked OOVs in terms of word frequency and by excluding self-describing words and words already received global acceptance. To collect candidate documents for a given NCW, we use three kinds of web references, Google, Urbandictionary, and Wikipedia, as a search engine, a user-contributed Internet jargon dictionary, and a user-contributed encyclopedia. When inferring topics of candidate documents of Urbandic-tionary and Wikipedia, we use 20 vocabulary words surrounding a given NCW. When extracting candidate documents from Google, we reference the snippets of its result.

Overall accuracy Given 200 NCWs of each test dataset, we collect candidate documents and apply a PLSA model to infer their topics. Then we manually read through three recommendations and rate each with three options: not relevant, somewhat relevant, and relevant. We rate a candidate document for a NCW as rele-vant only when the document contains the sentences to describe the NCW. Finally we accept candidates with relevant as answers. Figure 2(a) shows that, given 200 NCWs from D-my-TST, our approach finds answers of 85, 127, and 139 NCWs within top-1, top-2, and top-3 recommendations, respectively. For 200 NCWs from D-tw-TST we provide proper documents for 104, 146, and 161 within top-1, top-2, and top-3 recommendations as shown in Figure 2(b).

Comparison against baselines We compare our approach against three baselines where users reference only one web service, Google, Urbandictionary, or Wikipedia. For this, we also read through and rate all candidates with the same rating policy previously used. Figure 2(a) and 2(b) show the comparison result with D-my-TST and D-tw-TST. Two figures imply that our approach outperforms three baselines. In many cases, Google provides direct web links in the first result and presents the explanations of given words below. While Urbandictionary deals with various Internet jargon and newly coined named entities, we frequently observe that the qualities of the explanations are irregular. The documents in Wikipedia are mainly well-written but it misses many NCWs.

Original positions of answers In Figure 2(c), we plot the original position of answers at a web reference (i.e., Google, Urbandictionary, or Wikipedia). Here the answers are the documents which properly describe the given NCWs as well as position at between the first and the third places in our system. This figure implies that while about 86.4% (84.1%) of answers of 200 NCWs from D-my-TST (from D-tw-TST) were originally placed within the third position, about 13.6% (15.9%) of answers were placed below them, which users may not even look at.

5. Conclusion and Future Work

In this paper, we investigated how to find the meaning of a newly coined word (NCW) from a set of candidates from multiple web references. The approach is based on PLSA topic model and its overall process involves: (a) building a topic model with learning documents, (b) inferring the topic of a document containing a given NCW, (c) collecting a set of candidate documents for the given NCW and inferring the joint probability between the topic and each candidate, and (d) recommending the most relevant candidate by choosing the candidate with the maximum joint probability. We validate our proposed approach with two datasets crawled from MySpace and Twitter. To build topic models for two learning datasets, we introduced 50 and 70 which were measured based on perplexity information as the numbers of topics for MySpace and Twitter datasets, respectively. From each test dataset, we prepared 200 NCWs after choosing infrequent OOV words and excluding globally accepted words from them. Based on our manual inspection, our approach found the relevant meanings for 200 NCWs of MySpace dataset (200 NCWs of Twitter dataset) with the accuracy of 69.5% (80.5%) with three recommendation from our approach.

For future work, we are interested in using other topic models. We have used PLSA to infer the topic of a document but we can easily extend our approach to use other state-of-the-art approach on topic models, such as Latent Dirichlet Allocation (LDA) [4] and N-gram topic model [5]. Since they have different properties, such as Dirichlet prior topic distribution in LDA and N-gram based learning and inferring in N-gram topic model, we can compare the performance affected by other topic models with real-world datasets. Building a topic model is a time-consuming task especially when a learning dataset is huge. Thus, we are implementing a PLSA topic model over cloud computing environment [6], where one can process a heavy task with parallel subtasks using as a service Internet-based connected computing resources. We expect that to build a topic model over this environment allows service providers easily to adopt our approach to their data. We used three web references, Google, Urbandictionary, and Wikipedia, to collect candidate documents to explain a given NCW. However we failed to find the meanings of some NCWs, e.g., pse meaning Adobe Photoshop Elements, when they are very recently coined or a few of users are using. We will see the performance of our approach when we introduce more candidates from additional references.

References

[1] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, in: In Proc. 15th confernce on Uncertainty in Artifi-cial Intelligence, 1999, pp. 289–296.

[2] E. Agichtein, E. Brill, S. Dumais, R. Ragno, Learning user interaction models for predicting web search result preferences, in: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, 2006, pp. 3–10. doi:http://doi.acm.org/10.1145/1148170.1148175.

[3] T. Joachims, F. Radlinski, Search engines that learn from implicit feedback, Computer 40 (8) (2007) 34–40. doi:http://dx.doi.org/10.1109/MC.2007.289.

[4] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022. doi:http://doi.acm.org/10.1145/1362782.1362785.

[5] X. W. Xuerui Wang, Andrew McCallum, Topical n-grams: Phrase and topic discovery, with an application to information retrieval, Data Mining, IEEE International Conference on 0 (2007) 697–702.

[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, M. Zaharia, Above the clouds: A berkeley view of cloud computing (Feb 2009).