Beyond Bag-of-Words: A New Distance Metric for Keywords Extraction and Clustering

Shengqi Zhu
sqzhu@cs.wisc.edu

Abstract

Bag-of-Words (BoW) is a widely used model in a variety tasks in Natural Language Processing (NLP). However, this model does not consider any relations between words in the bag, which will bring about multiple problems in some NLP aspects. In this project, I proposed a framework for calculating pair-wise word relations within a bag, using both deterministic Wordnet database and stochastic context information. The final relation matrix could be viewed as both state transition matrix and inner product matrix, which will be helpful for both keywords abstraction and clustering tasks commonly seen in meta-search engines.

Introduction

Bag-of-Words (BoW) is a successful model widely used in Natural Language Processing (NLP). It assumes every word in a document independent with each other, regardless of ordering, context, etc. However, this assumption is sometimes oversimplified. One reason is that BoW does not take into consider any relations between words within a bag. This relationship is sometimes important to specific tasks, especially when the words in the bag have a strong semantic correlation with each other.

Let's take clustering as an example. If we have three different documents: an empty document, a document contains word "cat", and a document contains word "kitty". They could form a dictionary of [cat, kitty]. Under BoW, these three documents could be represented in two-dimensional space as $[0, 0]$, $[1, 0]$, and $[0, 1]$, respectively. Figure 1-left shows this representation in a cartesian coordinates. As we can see, it is extremely difficult to cluster them into two groups under such a representation. But intuitively, we know that "cat" and "kitty" are almost same in terms of semantics, hence having a very close relationship. Therefore, if we no longer hold the words in the bag as orthogonal, it is natural for us to "bend" the axis and thus easily clustering them into two different groups. This is shown in Figure 1-right.

The goal of my project is thus trying to find a framework which could quantify such relations within a bag. Moreover, I have also applied such framework to two common meta-search tasks: Keywords extraction and clustering, to show the improvement caused by accounting for word relationships.

Relation Representation

NLP usually relies on statistical properties extracting directly from data. Nevertheless, due to the limitation of meta-search results, data are usually not large enough to be fully confident with. Therefore, I used two different sources of information to determine the relations between words.

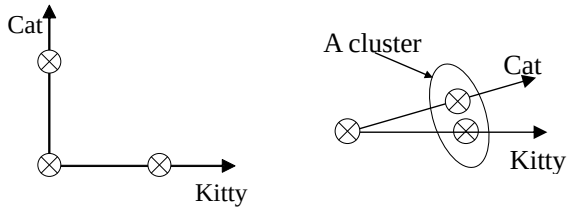

Figure 1: An example of clustering three documents containing two words

One of the sources is the Wordnet(Fellbaum 1998). Wordnet is a large lexical database of English language, where different words are related with each other in terms of taxonomy and form a hierarchical structure. Therefore, a path in this lexical forest could somehow represent the relations between words it connected with. There are in fact several papers studying the similarity metrics using Wordnet, such as (Hirst and St Onge 1998), (Leacock and Chodorow 1998), and (Banerjee and Pedersen 2003). In this project, I adopted the idea from (Wu and Palmer 1994), and implementation from nltk(Loper and Bird 2002). Wu's method has the advantage of being both simple and superior to other methods. Their similarity depends on the most specific ancestor node of two words in terms of taxonomy.(Loper and Bird 2002)

Another source I used is the context information. It is based on the assumption that words appear in close locations actually have close relationships. More specifically, if we put a window on one document we collected, all words co-occur in this window should have strong relations with each other. If we slide the window from the beginning of each document to the end, we may get all possible pair-wise word relations. This idea is similar to (Cao, Nie, and Bai 2007).

We could get a relation matrix from each of these two sources in the form of Eq (1).

$$\begin{bmatrix} rel(w_1,w_1) & rel(w_1,w_2) & \cdots & rel(w_1,w_n) \\ rel(w_2,w_1) & rel(w_2,w_2) & \cdots & rel(w_2,w_n) \\ \vdots & \vdots & \ddots & \vdots \\ rel(w_n,w_1) & rel(w_n,w_2) & \cdots & rel(w_n,w_n) \end{bmatrix} \quad (1)$$

By properly normalization and combining them with suitable coefficients, we end up with one unified matrix. Unfor-

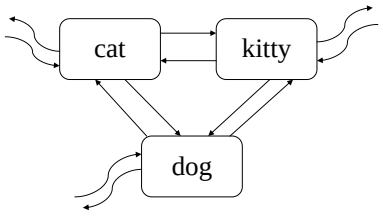

Figure 2: An example of generating a transition graph from relation matrix

tunately, the optimization of coefficients was not studied in this project and is left to be improved in the future.

Relation Interpretation

Eq (1) defines a new metric for relations between words within a bag. However, this matrix needs additional interpretation before we can apply it to real problems. In this project, I tried and found two different interpretation methods. They are corresponding with two applications.

Interpret from State Transition

(Cao, Nie, and Bai 2007) suggest that if we view each word as a state and each pair-wise relation as the probability to transit between states, then we could reconstruct a transition graph. This could be shown in Figure 2.

We are especially interested in the steady states properties of such transition model. Given this graphical interpretation, we could easily apply a Google Pagerank algorithm(Zhu 2009) and acquire a stable distribution. The "pagerank" of each word somehow represents the importance of each word in the bag. By sorting the pagerank value and returning word with top rank, we could actually extracting important keywords from documents.

Interpret from Inner Product

We could also view the matrix in Eq (1) as an inner product kernel matrix. Each element $rel(w_i, w_j)$ represents the inner product of two word vectors $\langle w_i, w_j \rangle$. Of course, we should first preprocess such matrix to make it conform to inner product definitions, such as symmetry, etc. Then, by applying Gram-Schmidt Orthonomalization Process(Golub and Van Loan 1996), we could transit such word vectors into a set of orthonormal basis. This is exactly the "bending" process shown in Figure 1. By representing each word vector under a set of orthonormal basis, we could now calculate real cartesian coordinates. These coordinates take into account the relations between words and are therefore suitable for clustering.

Results

Keywords Extraction

To test the effectiveness of keywords extraction tasks, I submitted different search requests through my system, along with the clusty website(Vivisimo 2005), and compared the first and the last several results after pagerank algorithms. Table 1 shows the result for search request "lemon", which suggests my result is very similar to the result from Clusty. But my keywords list used only description from top 50 pages, far fewer than what Clusty used. This is in fact the achievements from using Wordnet as prior information. Besides, the first few keywords are actually meaningful words while the last few are meaningless, indicating the effectiveness of pagerank algorithm on keywords extraction task.

Table 1: Keywords extracted from search query "lemon"

Clusty	My system	
	First 6	Last 6
Lemon Law	Lemon	Completely
Recipe	Juice	St
Lemon Tree	Law	Ever
Cake	Tree	Utilize
Picture	Fruit	Non
California	State	Anything

Clustering

To test the effectiveness of clustering, I made search requests of four different keywords and asked my program to cluster these results into two groups. The four different keywords I picked were carefully chosen so that they could fall into two semantic groups. For example, "dog, cat, desk, chair", the first two and the last two could fall in one group respectively. I then compared my results using relation matrix with traditional method that does not count on that information. For simplicity, the clustering algorithm is k-means. The result is shown in Table 2. Here, "without keywords" means the searching request keywords are removed from the dictionary for higher difficulty. The number "x/y" means getting x correct results out of all y results. The result shows successful clustering using relation matrix over traditional methods. A higher accuracy rate could be attributed to the use of prior knowledge (Wordnet) and statistical knowledge (Context) between words in a relatively small sample size (50 websites for each keyword).

Table 2: Clustering testing result

Keywords	Relation	Tradition
Dog, cat, desk, chair (with keywords)	103/200	154/200
Dog, cat, desk, chair (without keywords)	112/200	157/200
book, read, pet, dog (without keywords)	140/200	153/200

I have to mention that these experiments have been done more times, but due to the limitation of space, those results cannot be shown here.

Conclusion

Relation Matrix partly overcomes the shortcomings of BoW model, and its application on keywords extraction and clustering shows a promising effectiveness in NLP tasks.

References

Banerjee, S., and Pedersen, T. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 805–810.

Cao, G.; Nie, J.-Y.; and Bai, J. 2007. Using markov chains to exploit word relationships in information retrieval. In Evans, D.; Furui, S.; and Soul-Dupuy, C., eds., *RIAO*. CID.

Fellbaum, C. 1998. *WordNet: An Electronical Lexical Database*. Cambridge, MA: The MIT Press.

Golub, G. H., and Van Loan, C. F. 1996. *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press.

Hirst, G., and St Onge, D. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Fellbaum, C., ed., *WordNet: An Electronic Lexical Database*, 305–332. MIT Press.

Leacock, C., and Chodorow, M. 1998. Combining local context with wordnet similarity for word sense identification. In *WordNet: A Lexical Reference System and its Application*.

Loper, E., and Bird, S. 2002. Nltk: The natural language toolkit.

Vivisimo, I. 2005. Clusty. http://www.clusty.com.

Wu, Z., and Palmer, M. 1994. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, 133 –138.

Zhu, J. 2009. Lecture notes for cs 769. Technical report, University of Wisconsin - Madison Computer Sciences Department.