Visualization of Relation and Generality of Words in Search Result

Hirokawa, Sachio
Research Institute for Information Technology, Kyushu University

Yin, Chengjiu
Division for Arts and Science, Kyushu University

Nakae, Hiroto
Graduate School of Information Science and Electrical Engineering, Kyushu University

出版情報：Proceedings of Asian Conference on Information Systems 2014 : ACIS 2014, pp.90-101, 2014. The Institute of Electrical Engineers of Japan(IEEJ)
バージョン：
権利関係：

Visualization of Relation and Generality of Words in Search Result

Sachio Hirokawa[†] **Brendan Flanagan**[‡] **Chengjiu Yin**[*] **Hiroto Nakae**[‡]
[†]Research Institute for Information Technology, Kyushu University
hirokawa@cc.kyushu-u.ac.jp
[‡]Graduate School of Information Science and Electrical Engineering, Kyushu University
[*]Division for Arts and Science, Kyushu University

Abstract

Visualization of characteristic words is effective and useful to interpret the search result of a query. Those words are useful not only to interpret but also to shrink and expand the search. The position of a related word is determined by the strength of relation between the word and the query. The strength of relation of words is not necessarily influenced by the generality of a word. A general word and a restricted word might have the same similarity to the query. The present paper propose a method to integrate the strength of relation and the generality of a word by applying the authors method of bootstrapping.

Keywords: Bootstrap,MindMap,Spanning Tree, Topic Drift

1 Introduction

Search engine is useful and crucial to find information from the huge amount of data on the Web. However, the search result heavily depends on the choice of the query word. An ambiguous query word may yield unrelated search result. Too specific word may returns no result. On the other hand, too general word returns so many results that we have to narrow the search with appropriate words. To solve these problem, there are many researches of query expansion and recommendation of related words. The present paper consider the problem with two viewpoints, i.e., the similarity of words and the generality of words. Similar words of a word can be obtained by co-occurrence analysis of words within the search result. Related words that have a specific meaning or that are used in specific contexts may be extracted from the search result. However, obtaining more general words requires more information that cannot be found in the search result. The documents have to be expanded to cover more general words. The present paper introduce a method to integrate the similarity and the generality of words using the spanning tree and the bootstrapping.

Extraction of characteristic words and their relation of the search result have been attracted many researchers as a main subject in text mining. Mihalcea and Tarau [5] and Palshikar [7] analysed the network of words constructed from the search result. Mihalcea and Tarau [5] proposed the TextRank method to evaluate the score of words by applying HITS algorithm. Pantel and Pennacchiotti [6] proposed the Espresso-style bootstrapping method to extract important information from the viewpoints of summarization.

These researches are independent. Radev and Mihalcea [8] and Komachi et al. [3] pointed out that they can be considered very similar as analysis of co-occurrence graph of words. These methods analyse bipartite graph that consists of two kinds of nodes, i.e., words and documents or words and sentences. The problem is to extract tightly connected subgraphs. Iterated traverse from one type of notes to another type of nodes guides us to the characteristic words or important sentences. It is often the case that the documents and the words retrieved by iterated traverse tend to have little connection to the original query we sent at the beginning. This phenomenon is known as semantic drift or topic drift.

The present paper applies the restricted bootstrapping method [12; 1] to evaluate the generality level of words with respect a query. Spanning tree and mind map visualization are proposed by extending the method [2]. The proposed graph is a directed graph whose nodes are related words of the query. The root node is the most general words related to the query. The level of generality of words are evaluated by comparing the query. General words are drawn with red gradation. Less general words are drawn with blue gradation.

2 Formulation of Generality Level by Bootstrapping

The search result of a query may contain general words. The score of those words are not very high if we use appropriate measure such as tf*idf. However, when we repeat bootstrapping process, the number of such document that contain them gradually increases and those words gain high score. The first author proposed the restricted bootstrapping method in [12; 1], where only top k words are used to retrieve the documents. Thus the propagation of the effect of the general words are restrained and the semantic drift can be prevented. If the restriction is tight, i.e., k is small, we obtain less general words. If k is large, then we obtain more general words. On the other hand, the ranking of the same word decreases as we increase k.

The process of bootsrapping is described as the following algorithm BS. The input of BS is a ranked list U of words and an integer k to determines the bootstrap restriction. We use the set $top(k, U)$ of the top k words in U for OR-search and obtain the search result $word(doc(X))$. We iterate this process until the set of the top k words does not change. In the algorithm, $doc(X)$ represents the set of documents obtained by OR-search of the words in X and $word(doc(X))$ represent the set of all words in $doc(X)$.

```
BS(U,k) {
  X = top(k,U)
  W = {}
  i = 0
  while(true){
    Wi = word(doc(X))
    W = top(k,Wi)
    last if W == X
    i = i+1
    X = top(W,k)
  }
  return W
}
```

Figure 1. Restricted Bootstrap Algorithm

At the moment, the authors do not have a proof that the bootstrapping converges at all, i.e. that $W == X$ will hold eventually. The process might lead to a cicle. However, the detection of a cycle is easily realized. Thus, the process can be terminated by a small modification of the bootstrapping algorithm. In our experience with real samples, we observed that $W == X$ holds in a few steps.

The generality level of a word with respect to a query q is defined as follows. Let $W(q, k)$ be the ranked list of words at the stage k obtained by BS. Let $rank(w, W(q, k))$ be the rank of a word w in W(q,k). The score vector $V(w, M)$ of a word w is represented with list of M ranks of w. This vector represents the generality level of a word. Let $V(u, M) = (r_1, r_2, \cdots, r_M)$ and $V(v, M) = (s_1, s_2, \cdots, s_M)$ be the rank vectors of u and v, respectively. We order u and v according to the lexicographic order for the rank vectors. We say u is more general than v if $V(u, M) < V(v, M)$. The generality level $gen(u)$ of a word u is defined by $gen(u) = rank(u)/M$, where $rank(u)$ is the ranking of u and is defined by $rank(u) = \#\{v|V(v, M) < V(u, M)\} + 1$.

We realized the algorithm using a general search engine GETA [1]. The default score of words and the ranking of words of GETA are similar to that of the SMART system [9]. The BS algorithm can be applied to other measurement of words. The related words obtained may vary depending the score. Comparison and evaluation of the effect of word sore is left as a further work.

3 Change of Rank in Bootsrapping Process

Figure 3 shows the list of top 10 words at the bootstrapping stage k with respect the initial query "wsd" (word sense disambiguation). The words are lined according to their score in descending order. The initial query "wsd" is in the top position at the stage $k = 1$. However, the rank of the word declines to $1, 2, 3, 4, 4, 4, 5, 5, 9, 10$ as k increases. More general words appear above the word "wsd".

Table 1 is the listing of words of Table 3 sorted by descending order with respect to lexicographic order of the rank vector. The first number of the rank vector displays the rank of a word at $k = 10$. The last number displays the rank of the word at $k = 1$. The number shown in the table is the rank of a word at the level k subtracted by 1. For example, we see that the

[1] Generic Engine for Transposable Association (GETA), http://geta.ex.nii.ac.jp

rank vector of the word "sense" is 3311111142. It means that the word "sense" is in the 3rd rank at $k = 1$, in 5-th rank at $k = 2$ and in 2nd rank for $k = 3, \cdots, 8$. "word" is the most general word and "namesak" is the least general word with respect to the query "wsd". The 9 words above "wsd" are more general than "wsd". The other words below "wsd" occur in the search result and can be considered less general or special to "wsd".

w_i	rank vector of w_i
word	0087876799
disambigu	1100000001
lexic	2299989999
sens	3311111142
ambigu	4422652329
co-occurr	5599999999
english	6999999999
lexicon	7699999999
wordnet	8733224437
wsd	9844333210
thesauri	9955449994
unambigu	9966599995
polysem	9978765559
remot	9999997699
abbrevi	9999998869
wep	9999999979
name	9999999989
world-set	9999999993
decomposition-bas	9999999996
semcor	9999999998
n1	9999999999
noun	9999999999
namesak	9999999999

Table 1. Rank Vector

4 Generation of Spanning Tree and Mind Map

Let $W = \{w[0], w[1], ..., w[L]\}$ be the ranked list of related words given a query word w obtained by restricted bootstrapping. We construct a tree with the root $w[0]$. Imagine that the tree is constructed for the set of words $W_i = \{w[0], w[1], ..., w[i-1]\}$. The i-th word w_i is connected from the word $w[j]$ in W_i which has the highest similarity with $w[i]$.

The nodes of the tree are labeled with $w[0], w[1], ..., w[L]$. The generality level of a word is shown with gradation color. The more general words than the query word w is colored red. The more general words are colored with deep red. The words below the query word w with respect to the rank vector are colored blue if they are in the search result of w. The word is not colored if it is not in the search result even if it is lower than w. The query word w is circled with red line.

```
SpanTree(W) {
  for(i=1;i<L;i++){
    from = 0; maxsim = 0;
    for(j=0;j<i;j++){
      sim = similarity(wi,wj);
      if (sim>maxsim) {
        from = j;
        maxsim = sim;
      }
      next if maxsim < threshold;
      link(w[from],w[i])
    }
  }
}
```

Figure 2. Spanning Tree Generation Algorithm

5 Example of Spanning Tree

The authors constructed a prototype system that generates a spanning tree given a query word. We used 45,719 abstracts of scientific articles in information retrieval and text mining. They are collected from 16 international conferences and 5 journals. We firstly constructed a search engine of the data set and constructed the tree generation system on the search engine.

Figure 4 shows the spanning tree for the query "wsd". Figure 5 is the same graph displayed as a mind map. The two numbers below each word displays the number of documents that contain the word and the generality level among the related words.

In Figure 4, the sub-tree below the node "wsd" represents the characteristic words that appear in the search result of "wsd". All the words in the sub-tree are colored blue and can be recognized as specific words related to "wsd". The two words "disambigu" and "word" can be found by following the path upward from the word "wsd". It is convincing that the two words are general concept of "wsd" since "wsd" is an acronym of

"word sense disambiguation". There are other 7 words colored with red with gradation. Note that they are not in the sub-tree of "wsd". Those words are obtained as the result of bootstrapping and are more general compared to "wsd". They are not necessarily tightly connected to "wsd". This spanning tree explains how topic drift happens in bootstrapping.

6 Conclusion and Further Work

Restricted bootstrapping continues the search of documents with top k words and the extraction of the feature words. The set of the top k keywords converges to a fixed set after a number of iterations. When the restriction is weaken, i.e., k is large, new key words which did not appear before appear at the high rank among the feature words. As a result, the words that appeared before lose the hight position in the ranking. Top M words in each process of $k = 1, \cdots, N$ are accumulated as the related words of the initial query word. Each related word is represented as a rank vector. The present paper proposed a formulation of generality level based on the rank vectors. Then generation algorithm of spanning tree and mind map are proposed using the ranked list of related words. An example of spanning tree is shown using the abstracts of scientific articles. It is confirmed that general words can be found in the tree. Those words do not necessarily appear in the search results of the initial query. Quantitative evaluation is necessary to confirm the effectiveness of proposed method.

Acknowledgment

The authors apreciate valuable and constructive comments of reviwers. This work was supported by JSPS KAKENHI Grant Number 25750084 and 24500176.

References

[1] S. Hirokawa, Feature Extraction using Restricted Bootstrapping, Proc. 11th International Conference on Computer and Information Science(ICIS),pp.283-288, 2012

[2] S. Hirokawa, B. Flanagan, T. Suzuki, C. Yin, Learning Winespeak from Mind Map of Wine Blogs, in S. Yamamoto (Ed.): Proc. HIMI 2014, Part II, LNCS 8522, pp. 383-393, 2014

[3] M. Komachi, T. Kudo, M. Shimbo, Y. Matsumoto, Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms, Proc. EMNLP 2008,pp. 1011-1020,2008

[4] T. McIntosh, J.R. Curran, Reducing semantic drift with bagging and distributional similarity, Proc. ACL-IJCNLP 2009, pp. 396-404, 2009

[5] R. Mihalcea, P. Tarau, TextRank:Bringing Order into Texts, In Proceedings of EMNLP'2004. pp.404–411, 2004

[6] P. Pantel, M. Pennacchiotti, Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations, Proc. 44th ACL, pp.113-120, 2006

[7] G. K. Palshikar, Keyword Extraction from a Single Document Using Centrality Measures, Springer LNCS 4815, pp. 503–510, 2007

[8] D. Radev, R. Mihalcea, Network and Natural Language Processing, AI Magazine, pp.16-28, 2008

[9] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983

[10] I.Suzuki, K. Hara, M. Shimbo, Y. Matsumoto, A graph-based approach for biomedical thesaurus expansion, Proc. International Conference on Information and Knowledge Management, pp. 79-82,2009

[11] X. Yan, R.Y.K. Lau, D. Song, X. Li, J. Ma, Towards a Semantic Granularity Model for Domain-specific Information Retrieval, ACM Transactions on Information Systems Vol. 29, No.3, pp.1-46, 2011

[12] J. Zeng, T. Sakai, B. Flanagan, S. Hirokawa, Extraction of Feature Words with the Same Generality Level as Query using Restricted Bootstrapping, Proc. IEEE 14th International Conference on Commerce and Enterprise Computing, pp. 171-176, 2012

k	loop	ranked words (top 10)
1	1	*wsd disambigu sens world-set thesauri unambigu decomposition-bas wordnet semcor n1
2	2	disambigu *wsd ambigu wordnet sens polysem abbrevi wep name namesak
3	2	disambigu sens *wsd ambigu wordnet polysem remot word abbrevi noun
4	3	disambigu sens ambigu *wsd wordnet polysem word remot abbrevi noun
5	3	disambigu sens wordnet *wsd thesauri ambigu polysem word lexic remot
6	3	disambigu sens wordnet *wsd thesauri unambigu ambigu polysem word lexic
7	3	disambigu sens ambigu wordnet *wsd thesauri unambigu word polysem lexic
8	3	disambigu sens ambigu wordnet *wsd thesauri unambigu polysem word lexic
9	5	word disambigu lexic sens ambigu co-occurr lexicon wordnet *wsd english
10	4	word disambigu lexic sens ambigu co-occurr english lexicon wordnet *wsd

Figure 3. Top 10 words at Bootstrapping Stage k

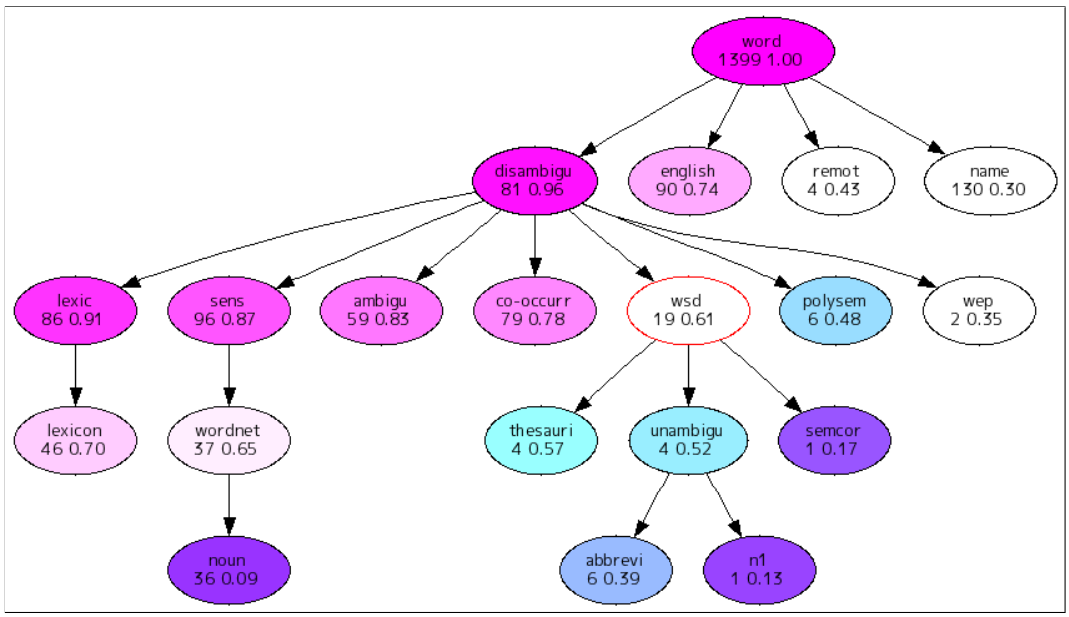

Figure 4. Spanning Tree of "wsd"

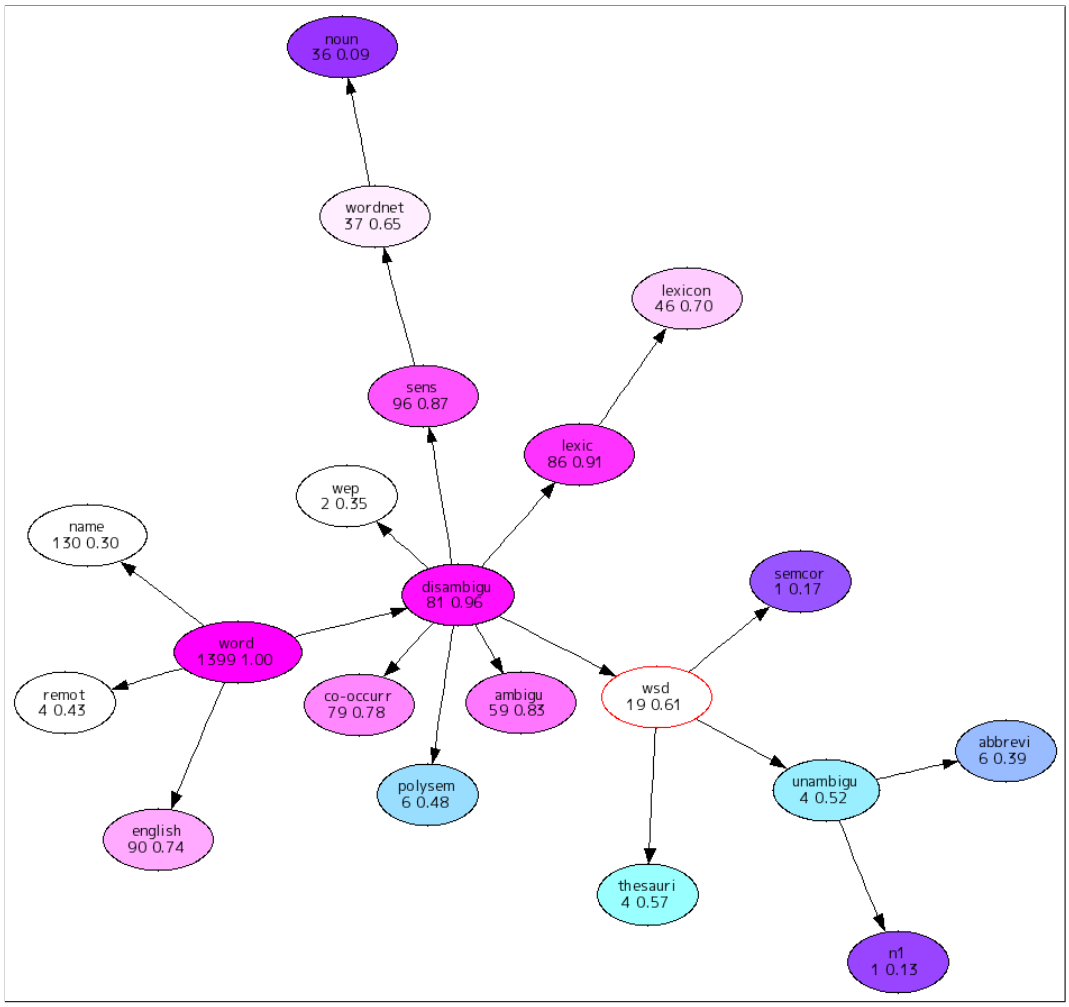

Figure 5. MindMap of “wsd”