Combining Classification with Clustering for Web Person Disambiguation

Jian Xu, Qin Lu, Zhengzhong Liu
Department of Computing, The Hong Kong Polytechnic University, Hong Kong
{csjxu, csluqin, hector.liu}@comp.polyu.edu.hk

ABSTRACT

Web Person Disambiguation is often conducted through clustering web documents to identify different namesakes for a given name. This paper presents a new key-phrased clustering method combined with a second step re-classification to identify outliers to improve cluster performance. For document clustering, the hierarchical agglomerative approach is conducted based on the vector space model which uses key phrases as the main feature. Outliers of cluster results are then identified through a centroids-based method. The outliers are then reclassified by the SVM classifier into the more appropriate clusters using a key phrase-based string kernel model as its feature space. The re-classification uses the clustering result in the first step as its training data so as to avoid the use of separate training data required by most classification algorithms. Experiments conducted on the WePS-2 dataset show that the algorithm based on key phrases is effective in improving the WPD performance.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering; I.2.7 [**Artificial Intelligence**]: Natural Language Processing - Text analysis.

General Terms: Algorithms, Experimentation

Keywords: Web person disambiguation, key phrase, string kernel, SVM

1. INTRODUCTION

Web Person Disambiguation (WPD) aims to identify the different namesakes for a given name [1]. It normally involves two parts. The first part uses clustering to group different namesakes and the second part extracts the actual attributes of each namesake in each cluster. This work focuses only on the first part of WPD. Most of the previous researches tend to use a rich set of features such as, tokens, named entities, URL, and snippets, etc. [2, 4, 6]. However, more types of features may also introduce more noises and more resource may be needed including clustering time.

This paper presents a novel WPD algorithm which uses only key phrases as it feature in clustering and also introduces a re-classification step for outliers to improve clustering results. The use of key phrases is based on the hypothesis that key phrases are better semantic representations of namesakes [5]. The algorithm uses the key phrases in both steps. In the first step, key phrases are used as the single feature in VSM for clustering. In the second step, wrongly clustered outlier documents are first detected by a centroid-based method and they are then re-classified by the SVM algorithm using the string kernel model based on key phrases which has a larger granularity than character or word based methods [3, 8].

WWW 2012 Companion, April 16-20, 2012*,* Lyon, France.
ACM 978-1-4503-1230-1/12/04.

2. ALGORITHM DESIGN

The proposed algorithm involves four components. The 1st component extracts key phrases associated with named entities using Wikipedia anchor text. The 2nd component uses the HAC algorithm to find the different namesake clusters. The 3rd component uses the centroid approach to identify the outliers in each cluster which are considered wrongly clustered. The 4th component identifies the more appropriate clusters for the outliers using each cluster as one class and using one-versus-one strategy to reclassify the outliers.

2.1 Key Phrase Extraction

In VSM-based clustering, different algorithms use different sets of features to represent a document including tokens, bigrams, and hyperlinks, etc. [4]. The selection of features directly affects the performance of the algorithms. This work chooses to use only key phrases as the features for clustering. To avoid manual annotation for the training data, Wikipedia's personal names articles are used as the training corpus and anchor text contained in these articles serve as the annotated key phrases. Anchor texts in Wikipedia are manually labeled by crowds, thus are semantically sound and reliable. In this work, the extraction algorithm uses the Naive Bayes (NB) learning strategy [10]. The extraction of the key phrases in the training phase is fully automatic and independent of WePS datasets.

2.2 HAC Clustering

For personal name type of document clustering, the hierarchical agglomerative approach (HAC) is used. Document similarities are computed through VSM which uses key phrases extracted in Section 2.1. The weight for a key phrase, denoted by k, considers both the commonly used *TF*IDF* and their Wikipedia link probabilities [9] as defined below:

$$W_k = \log(TF(k)+1)\times(\log IDF(k)+P_{link}(k))$$

where $TF(k)$ denotes the term frequency of k, $IDF(k)$ is the inverse document frequency of k, and $P_{link}(k)$ is the link probability of k.

2.3 Outlier Identification and Reclassification

As clustering may produce some outlier documents which may belong to a different namesake cluster, the proposed algorithm remedies this by first identifying the outliers through a centroids-based approach based on the observations that the distances of the outliers to their own cluster centroids are farther than that of other clusters. Then the outliers are re-classified into a different cluster.

The classifier uses a key phrase-based string kernel model [3, 8] for comparison. Any two documents, represented by their key phrase sequences s, and t, are compared through the key phrase subsequences as features. To control the feature space, the string kernel has a parameter n which denotes the length of subsequences to be considered. Then, the similarity measure, denoted by $K_n(s,t)$, is defined as:

$$K_n(s,t) = \sum_{u\in\sum^n} \left\langle \phi_u(s)\cdot\phi_u(t)\right\rangle = \sum_{u\in\sum^n}\ \sum_{i:u=s[i]}\lambda^{l(i)}\ \sum_{j:u=t[j]}\lambda^{l(j)}$$

where λ is the decay factor that penalizes the non-continuous subsequences, $\sum^n$is the set of all key phrase subsequences of length *n*. *s[**i**]* and *t[**j**]* are subsequences of *s* and *t*, respectively. *l(**i**) and l(**j**)* are the lengths of *s[**i**]* and *t[**j**]*, respectively. To avoid enumeration of all subsequences in *s* and *t*, dynamic programming is used in this work [8].

To further reduce computation complexity, only distinct key phrases in the high ranked sentences are used. Sentence ranking is produced through a sentence to key phrase bipartite graph and then the HITS algorithm [7] is used to rank the sentences.

3. EXPERIMENTS

The evaluation is done on the WePS2 dataset with 30 ambiguous names. Each name has around 150 documents. For key phrase extraction, the training data are from the Wikipedia person name articles identified using the persona name list in DBpedia[1].

The performance is evaluated by B-Cubed and Purity scores in Table 1 and Table 2, respectively. In both tables, two F-measures are also used, one giving equal weighting to precision and recall (α=0.5) and the other giving higher weighting to recall (α=0.2). Let $A_{CLUSTER}$ denotes the proposed key phrase-based clustering algorithm and A_{COMB} refers to the two step algorithm with re-classification. For re-classifying the outliers, the tool LIBSVM[2] is used. The performances of $A_{CLUSTER}$ and A_{COMB} are compared to the top 3 systems in WePS2 [2, 4, 6].

Table 1. Comparison using B-Cubed scores

Runs	Macro-averaged Scores			
	F-measures		B-Cubed	
	α= 0.5	α= 0.2	Pre.	Rec.
T1: PolyUHK	82	80	87	79
T2: UVA_1	81	80	85	80
T3: ITC_UT_1	81	76	93	73
A $\boldsymbol{A_{CLUSTER}}$	**84**	**85**	82	**86**
$\boldsymbol{A_{COMB}}$	**84**	**85**	**83**	**86**

Table 2. Comparison using Purity scores

Runs	Macro-averaged Scores			
	F-measures		Purity	
	α= 0.5	α= 0.2	Pur.	Inv_Pur.
T1: PolyUHK	**88**	87	91	86
T2: UVA_1	87	87	89	87
T3: ITC_UT_1	87	83	**95**	81
$\boldsymbol{A_{CLUSTER}}$ **&** $\boldsymbol{A_{COMB}}$	**88**	**89**	86	**90**

Table 1 and Table 2 show that $A_{CLUSTER}$ and A_{COMB} have the best results in terms of F-measure for both B-cubed scores and purity scores because the proposed algorithms have a good balance between precision and recall. The most obvious improvement is in recall reflected both by B-Cubed and inverse purity with gains of 7% and 4%, respectively. The reason that the proposed algorithm has better recall is that the key phrase based approach reduced the number of noise in the feature space making the features more distinctive. In terms of purity, the proposed algorithm However, A_{COMB} gained only 1% in B-cubed score compared to $A_{CLUSTER}$, and it has no improvement for purity measure. It should be pointed out that outlier detection and reclassification do not change the number of clusters. Thus this step can only improve the quality of memberships in each cluster. Further analysis show that in detecting outliers, 15 documents are identified and the precision is 93.33% (14/15) which is quite good. However, out of the 3,438 documents, 15 is not a significant number which explains the relative small improvement. As for purity, further error analysis shows that poor performance of the reclassification algorithm is due to unbalanced data in different clusters.

To validate the use of key phrase-based string kernel for SVM classifier, the A_{COMB} algorithm is compared to other kernels: RBF, Polynomial and Linear. Experiments show that the performance of A_{COMB} is marginally better than the other kernels in B-Cubed precision scores by 1 percent and they are basically the same in terms of purity scores.

4. CONCLUSIONS

This paper proposed a key-phrase based clustering algorithm with outlier reclassification for WPD. It investigates the use of key phrases as a single feature for clustering. The extraction of key phrases is trained by anchor text in Wikipedia so that no other annotation for training is needed. To further improve the clustering results, an attempt is made to use SVM classifier for outlier reclassification. Even though the precision of outlier identification is very good, further investigation is still needed to see how they can be used to improve the clustering results in terms of cluster numbers and outlier reclassification.

5. REFERENCES

[1] Artiles, J., Borthwick, A., Gonzalo, J., Sekine, S., and Amigo, E. 2010. WePS-3 evaluation campaign: overview of the web people search clustering and attribute extraction tasks. *CLEF 2010.*

[2] Balog, K., He, J., Hofmann, K., et al. The University of Amsterdam at WePS2. *WePS 2009.*

[3] Cancedda, N., Gaussier, E., Goutte, C. & Renders, J.M. 2003. Word Sequence Kernels. *Journal of Machine Learning Research*.

[4] Chen, Y., Lee, Y. M., Chu-Ren Huang. 2009. PolyUHK: A robust information extraction system for web personal names. *WePS 2009.*

[5] Hulth, A. and Megyesi, B. 2006. A study on automatically extracted keywords in text categorization. *CoLing/ACL 2006*. Sydney (2006).

[6] Ikeda, M., Ono, S., Sato, I., Yoshida, M., Nakagawa, H. 2009. Person name disambiguation on the web by two-stage clustering. *WePS 2009, 18th WWW Conference*.

[7] Kleinberg, J. 1998. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM SODA*.

[8] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C. 2002. Text classification using string kernels, *The Journal of Machine Learning Research*, 2, p.419-444.

[9] Milne, D. 2007. Computing semantic relatedness using Wikipedia link structure. In *Proc. NZ CSRSC'07*, Hamilton, New Zealand.

[10] Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C. and Nevill-Manning, C.G. 2000. *KEA: Practical Automatic Keyphrase Extraction*. Working Paper 00/5.

[1] http://wiki.dbpedia.org

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/