Aggregating Skip Bigrams into Key Phrase-based Vector Space Model 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 (WPD) is often done through clustering of web documents to identify the different namesakes for a given name. This paper presents a clustering algorithm using key phrases as the basic feature. However, key phrases are used in two different forms to represent the document as well context information surround the name mentions in a document. In using the vector space model, key phrases extracted from the documents are used as document representation. Context information of name mentions is represented by skip bigrams of the key phrase sequences surrounding the name mentions. The two components are then aggregated into the vector space model for clustering Experiments on the $WePS2$ datasets show that the proposed approach achieved comparable results with the top 1 system. It indicates that key phrases can be a very effective feature for WPD both at the document level and at the sentential level near the name mentions.

1 Introduction

Most of current search engines are not suited for web persons disambiguation because only pages related to the most popular persons will be easily identified. Web Persons disambiguation (WPD) targets at identifying the different namesakes for a given name (Artiles et al., 2010). Normally WPD involves two steps. The first step uses clustering methods to cluster different namesakes and the second step works on each cluster to extract the descriptive attributes of each namesake to form their profiles. This paper focuses on the clustering algorithms in WPD.

Most of the previous researches attempted to use a combination of different features such as, tokens, named entities, URL or title tokens, n-gram features, snippets and other features (Chen et al.,2009; Chong et al., 2010). Traditionally, document clustering based on a single representation space using the vector space model (VSM) is often the choice (Salton and McGill, 1983). However, how to find a good balance between the selection of a rich set of features and degradation performance due to more noise introduced is an important issue in VSM.

This paper presents a clustering algorithm based on using key phrases only. The use of key phrases is based on the hypothesis that key phrases, or sometimes referred to as topic words (Steyvers and Griffiths, 2007), are better semantic representations of documents (Anette Hulth, 2003). We also argue that the key phrases surrounding the name mentions can represent the context of the name mentions and thus should be considered as another feature. This is an important distinction on clustering for WPD compared to the purpose of other document clustering algorithms. In this paper, key phrases are thus used in two parts. In the first part, key phrases are used as the single feature to be represented by the VSM for clustering. In the second part, the key phrases in a sequential representation surrounding a name mention are identified using skip bigrams. Finally, the skip-bigrams are concatenated to the bag of key phrase model to serve as the aggregated key phrase-based clustering (AKPC) algorithm.

For key phrase extraction, a supervised learning algorithm is used and trained through the English Wikipedia personal article pages so as to avoid laborious manual annotation. To

incorporate the context information at sentential level into WPD, the name mentions in the document are first located and then the key phrases that surround the name mentions are extracted. These key phrases are arranged into sequences from which the skip bigrams are then extracted.

Different from the previous skip bigram statistics which considers pairs of words in a sentence order with arbitrary gaps (Lin and Och, 2004a) and compares sentence similarities through the overlapping skip bigrams, the skip bigrams in this paper are weighted by an exponentially decay factor of their full length in the sequence, hence emphasizing those occurrences of skip bigrams that has shorter skips (Xu et al., 2012). It is reasonable to assume that if two sentences are similar, they should have many overlapping skip bigrams, and the gaps in their shared skip bigrams should be similar as well. Besides, a different weighting scheme for skip bigrams in this paper is used. It combines the penalizing factor with the length of gaps, named skip distance (SD). The longer the skip distance is, the more discount will be given to the skip bigrams.

The rest of this paper is organized as follows. Section 2 describes the related works of web person disambiguation. Section 3 presents key phrase extraction algorithm. Section 4 describes the skip bigrams. Section 5 gives the performance evaluation of the aggregated key phrase-based clustering (AKPC) algorithm. Section 6 is the conclusion.

2 Related Work

Web Person Disambiguation, as a task was defined and contested in the WePS workshops 2007, 2009, 2010 (Artiles et al., 2007, 2009, 2010). In WePS workshops, both development data (including training data and golden answer) and testing data are provided. The searched results include snippets, ranking, document titles, their original URLs and HTML pages (Artiles et al., 2009).

Some harvested the tokens from the web pages external to the WePS development data (Chen et al.,2009; Han et al., 2009), and others used named entities (Popescu et al., 2007). Some algorithms used external resources such as Google 1T corpus and Wikipedia to tune the weighting metrics. For example, Chen et al. (2009) used the Google 1T 5-gram data to learn the bigram frequencies. Chong et al. (2010) used Wikipedia to find phrases in documents.

Key phrases give a semantic summarization of documents and are used in text clustering (Hammouda et al., 2005), text categorization (Hulth and Megyesi, 2006) and summarization (Litvak and Last, 2008). For key phrase extraction, supervised and unsupervised approaches are both commonly used. Wan and Xiao (2008) proposed the CollabRank approach which first clustered documents and then used the graph-based ranking algorithm for single document key phrase extraction. Zha (2002) applied the mutual reinforcement principle to extract key phrases from a sentence based on the HITS algorithm. Similarly, Liu et al. (2010) considered the word importance related to different topics when ranking key phrases. Li et al. (2010) proposed a semi-supervised approach by considering the phrase importance in the semantic network. Frank et al. (1999) and Witten et al. (2000) used the Naive Bayes approach to extract key phrases with known key phrases. Similarly, Xu et al. (2012) proposed to use the anchor texts in Wikipedia personal articles for key phrase extraction using the Naive Bayes approach.

Skip bigram statistics are initially used to evaluate machine translation. It measures the overlap between skip bigrams between a candidate translation and a set of reference translations (Lin and Och, 2004a). The skip bigram statistics uses the ordered subsequence of words as features for sentence representation in machine translation evaluation. It counts the matches between the candidate translation and a set of reference translations. However, there is no attempt to use key phrases to create skip bigrams for WPD.

3 Key Phrase Extraction

In VSM based clustering, different algorithms use different set of features to represent a document such as tokens, name entities (Popescu et al., 2007; Chen et al., 2009; Han et al., 2009).

The choice of features directly affects both the performance and the efficiency of their algorithms. Simple features can be more efficient, but may suffer from data sparseness issues. However, more features may also introduce more noise and degrade the performance and efficiency of the algorithm. This paper investigates the use of key phrase as the single feature for WPD. Key phrases are similar to topic words used in other applications as semantic representations (Steyvers and Griffiths, 2007). The difference is that key phrases are bigger in granularity, which can help reduce the dimensionality of the data representation. More importantly, key phrases are better representation of semantic units in documents. For example, the key phrase $World\ Cup$ denotes the international football competition, if it is split into $World$ and Cup, its semantical meaning would be altered.

Extraction of key phrases can take different approaches. If training data is available, some learning algorithms can be developed. However, annotation is needed to prepare for the training data which can be very time consuming. To avoid using manual annotation, we resort to using anchor text in Wikipedia as training data for key phrase extraction. Anchor texts in Wikipedia are manually labeled by crowds of contributors, thus are meaningful and reliable. Figure 1 is an excerpt of the Wikipedia personal name article for the American president *Abraham Lincoln*:

Abraham Lincoln /ˈeɪbrəhæm ˈlɪŋkən/ (February 12, 1809 – April 15, 1865) was the 16th President of the United States, serving from March 1861 until his assassination in April 1865. He successfully led his country through its greatest constitutional, military and moral crisis – the American Civil War – preserving the Union while ending slavery, and promoting economic and financial modernization. Reared in a poor family on the western frontier, Lincoln was mostly self-educated. He became a country lawyer, a Whig Party, Illinois state legislator in the 1830s, a one-term member of the United States House of Representatives in the 1840s, but he failed in two attempts to be elected to the United States Senate in the 1850s. After opposing the expansion of slavery in the United States in his campaign debates and speeches,[1] Lincoln secured the Republican Party nomination and was elected president in 1860.

Figure 1: Excerpt of a Wikipedia article

In this excerpt, $American\ Civil\ War$, $Whig\ Party$, $United\ States\ Senate$, $Illinois\ state\ legislator$, $Republican\ Party$ and other anchor texts can be used as key phrases. Using the Wikipedias personal names articles, key phrase extraction algorithm can then be employed to train the prediction model. In this work, the extraction algorithm uses the Naive Bayes (NB) learning strategy for training through the use of anchor texts in Wikipedia personal names articles to extract key phrases. The list of personal names in Wikipedia is first obtained from DBpedia[1] which are used to obtain the relevant Wikipedia personal articles. The NB algorithm creates the key phrase prediction model using the extracted key phrases during the training process. Similar to the supervised key phrase extraction approaches (Witten et al., 2000; Xu et al., 2012), our key phrase extraction is summarized as follows.

- Preprocessing: Clean the Wikipedia articles including html tags removal, text tokenization, lemmatization and case-folding;
- Anchor text extraction: Extract the anchor texts based on the embedded hyperlinks;
- Candidate phrase generation: Use ngram-based method to generate candidate phrases which can contain up to 3 words as a phrase. They cannot start and end with stop words;
- Annotation: Label the candidate phrases with anchor text as positive instances and others as negative instances;
- Feature value generation and discretization: Compute (1) candidate phrases TF*IDF values, and (2) the distance values by the number of words preceding the candidate phrases divided by the document length in words. If there are multiple candidate phrases in the same document, the value of its first appearance will be used;
- Classification: Use the Naive Bayes learning algorithm to produce the key phrase prediction model.

The NB classification for positive prediction is formally defined as:

$$P(yes|k) = \frac{Y}{Y+N} \times P_{tf*idf}(t|yes) \times P_{dist}(d|yes)$$

where k is a phrase, Y and N denote positive and negative instances. Positive instances are

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

those candidate phrases that are anchors in Wikipedia and negative ones are those candidate phrases which are not anchors. t is the discretized TF*IDF value and d refers to the discretized distance value.

4 Key Phrase-based Skip Bigrams

Skip-bigrams are pairs of key phrases in a sentence order with arbitrary gaps. They contain the sequential and order-sensitive information between two key phrases. Xu et al. (2012) extracted skip bigrams based on the words to measure sentence similarities. In this paper, we used the sequences of key phrases surrounding a name mention. To use the skip bigrams, the key phrase sequences are first extracted within a context window of the name mentions. Figure 3 shows the key phrases surrounding the mention of *Amanda Lentz*.

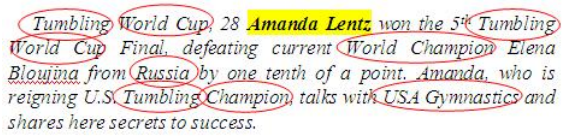

Figure 2: Key Phrases for person $Amanda\ Lentz$

In this short text, the key phrases in the red circles (their extraction will be described in Section 3). To find the skip bigrams, we first pinpoint the person name *Amanda Lentz*, find the key phrases surrounding this name mention by specifying the window size, and then create a key phrase sequence as follows:

tumbling world_cup ***amanda_lentz*** *tumbling world_cup world_champion russia tumbling champion usa_gymnastics*

From the above key phrase sequences, the skip bigrams are extracted. Without loss of generality, let us consider the following examples of key phrase sequences S_1 and S_2 around a name mention:

S_1=k_1 k_2 k_1 k_3 k_4 and S_2=k_2 k_1 k_4 k_5 k_4

where k_i denotes a key phrase. It can be used more than once in a key phrase sequence. Hence, S_1 has the following skip bigrams:

(k_1k_2, k_1k_1, k_1k_3, k_1k_4, k_2k_1, k_2k_3 , k_2k_4 , k_1k_3, k_1k_4, k_3k_4)

S_2 has the following skip bigrams:

(k_2k_1, k_2k_4, k_2k_5, k_2k_4, k_1k_4, k_1k_5 , k_1k_4 , k_4k_5, k_4k_4, k_5k_4)

In the key phrase sequence S_1, we have two repeated skip bigrams k_1k_4 and k_1k_3. In the sequence S_2, we have k_2k_4 and k_1k_4 repeated twice. In this case, the weight of the recurring skip bigrams will be increased. Now, the question remains of how to weigh the skip bigrams.

Given Ω as a finite key phrase set, let $S = k_1k_2\cdots k_{|S|}$ be a sequence of key phrases for a name mention, $k_i \in \Omega$ and $1 \leq i \leq |S|$. A skip bigram of S, denoted by u, is defined by an index set $I = (i_1, i_2)$ of S such that $1 \leq i_1 < i_2 \leq |S|$ and $u = S[I]$. The skip distance of $S[I]$, denoted by $l_u(I)$, is the skip distance of the first key phrase and the second key phrase of u in S, calculated by $i_2 - i_1 + 1$. For example, if S is the key phrase sequence of $k_1k_2k_1k_3k_4$ and $u = k_1k_4$, then there are two index sets, $I_1 = [3, 5]$ and $I_2 = [1, 5]$ such that $u = S[3, 5]$ and $u = S[1, 5]$, and the skip distances of $S[3, 5]$ and $S[1, 5]$ are 3 and 5, respectively. In case a name mention occurs multiple times in a document, the key phrase sequences for the name mentions are concatenated in their occurrence order to form one compound sequence. In the following discussions, S refers to the compound key phrase sequence if there are multiple name mentions.

The weight of a skip bigram u for a given S with all its possible occurrences, denoted by $\phi_u(S)$, is defined as:

$$\phi_u(S) = \sum_{I:u=S[I]} \lambda^{l_u(I)}$$

where λ is the decay factor, in the range of [0,1], that penalizes the longer skip distance $l_u(I)$ of skip bigrams. That is to say, the longer the skip distance is, more discount will be given to the skip bigrams.

By doing so, for the key phrase sequence S_1, the complete key phrase set is $\Omega = \{k_1, k_2, k_3, k_4\}$. The weights for the skip bigrams are listed in Table 1:

These extracted skip bigrams with their corresponding weights will be concatenated into the key phrase-based vector space model. Suppose two documents are represented by the

u	$\phi_u(S_1)$	u	$\phi_u(S_1)$
k_1k_2	λ^2	k_2k_1	λ^2
k_1k_1	λ^3	k_2k_3	λ^3
k_1k_3	$\lambda^4 + \lambda^2$	k_2k_4	λ^4
k_1k_4	$\lambda^5 + \lambda^3$	k_3k_4	λ^2

Table 1: Skip Bigrams and their Weights in S_1

key phrase vectors V_{S_1} and V_{S_2},

$V_{S_1} = (k_1, k_2, k_3, k_4)'$

$V_{S_2} = (k_1, k_2, k_4, k_5)'$

The symbol prime denotes the transpose of the row vectors. Once the skip bigrams are extracted, they are concatenated into their vector spaces and thus the V_{S_1} and V_{S_2} are expanded into

$$V_{S_1} = (k_1, k_2, k_3, k_4, k_1k_2, k_1k_1, k_1k_3, k_1k_4, k_2k_1, k_2k_3, k_2k_4, k_3k_4)'$$

$$V_{S_2} = (k_1, k_2, k_4, k_5, k_2k_1, k_2k_4, k_2k_5, k_1k_4, k_1k_5, k_4k_5, k_4k_4, k_5k_4)'$$

The V_{S_1} and V_{S_2} vectors are enriched after concatenation and if they share more overlapping skip bigrams with similar skip distances, the similarity between V_{S_1} and V_{S_2} will be increased.

5 Experiments

The evaluation of the algorithm is conducted using the test data of $WePS2$ workshop 2009 [2] which has 30 ambiguous names. Each ambiguous name has 150 search results from the various domains including US census, Programme Committee members for the annual meeting of ACL, and so on (Artiles et al., 2009).

Because the number of clusters is not known beforehand, the parameter configuration for clustering is of great importance for clustering web persons. In this paper, the $WePS1$ development data[3] is used to select the optimal threshold. This development data contains 47 ambiguous names. The number of clusters per name has a large variability from 1 to 91 different people sharing the name (Artiles et al., 2009). In the preprocessing step, the software Beautiful Soup[4] is used to clean the html texts and the OpenNLP tool[5] to tokenize cleaned texts.

[2]http://nlp.uned.es/weps/weps-2

[3]http://nlp.uned.es/weps/weps-1

[4]http://www.crummy.com/software/BeautifulSoup/

[5]http://incubator.apache.org/opennlp/

5.1 Key Phrase Extraction

For key phrase extraction, no training data from WePS is used. Instead, the training data are from the Wikipedia personal names articles. The personal names in Wikipedia are available from the DBpedia. 245,638 personal names are used in this paper with their corresponding Wikipedia articles. These persons come from different walks of life, thus providing a wide coverage of terms across different domains. Through the Wikipedia personal name article titles, the Wikipedia Miner tool[6] is used to obtain the anchor text within the article page. With the article pages as documents and the related key phrases (anchor texts), the key phrase prediction model is trained first. Then the key phrases in the WePS testing data are extracted. In case of overlapping key phrases, longer key phrases will be used. For example, $president\ of\ united\ states$ and $united\ states$ are both key phrases. But, when $president\ of\ united\ states$ appears in the context, it will be used even though both $present\ of\ united\ states$ and $united\ states$ are extracted simultaneously.

The key phrases extracted for the persons *AMANDA_LENTZ* and *BENJAMIN_SNYDER* are listed here as an example:

AMANDA_LENTZ*: IMDb, North Carolina, Literary Agents, published writers, High School, Family History, CCT Faculty, Campus Calendar, Women Soccer, World Cup, Trampoline, ⋯*

BENJAMIN_SNYDER*: Biography Summary, Artist, National Gallery of Canada, Fine Arts Museum, history of paintings, modern art work, University of Manitoba, Special Collections, portfolio gallery, ⋯*

It is quite obvious that above extracted key phrases are informative and useful for the WPD task. Compared to using topic words, the use of key phrases reduces the document dimension significantly, thus reducing runtime cost. When dealing with internet documents which can be in very large quantity, reduction of runtime cost can make the algorithms more practical.

[6]http://wikipedia-miner.cms.waikato.ac.nz/

5.2 Evaluation Metrics for WPD

The algorithm is evaluated by the purity, inverse purity scores, and B-Cubed precision and recall (Artiles et al., 2007, 2009). The purity measure is defined as

$$Purity = \sum_i \frac{C_i}{n} maxPre(C_i, L_j)$$

$$Pre(C_i, L_j) = \frac{C_i \cap L_j}{C_i}$$

where C_i denotes the i^{th} cluster produced by the system, L_j denotes the j^{th} manually annotated category and n the number of clustered documents. $Pre(C_i, L_j)$ refers to precision of a C_i for the category L_j. Inverse purity focuses on the cluster with the maximum recall for each category, defined by,

$$Inv\_Purity = \sum_i \frac{L_i}{n} maxPre(L_i, C_j)$$

To take into consideration of both precision and recall in evaluating clustering performance, the harmonic mean of both purity and inverse purity is defined as follows:

$$F = \frac{1}{\alpha \frac{1}{Purity} + (1-\alpha)\frac{1}{Inv\_Purity}}$$

where $\alpha = \{0.2, 0.5\}$ used in the WePS workshops (Artiles et al., 2009, 2010). If smaller α gives more importance to inverse purity, indicating a higher weight to recall. In the case of $\alpha = 0.5$, equal weighting is given to precision and recall.

B-Cubed metrics calculate the precision and recall related to each item in the clustering result. The B-Cubed precision (BEP) of one item represents the amount of items in the same cluster that belong to its category, whereas the B-Cubed recall (BER) represents how many items from its category belong to its cluster. They are,

$$BEP = Avg_e[Avg_{e' C(e) \cap C(e') \neq 0}[Mult.Pre(e, e')]]$$

$$BER = Avg_e[Avg_{e' L(e) \cap L(e') \neq 0}[MultRecall(e, e')]]$$

e and e' are two documents, $C(e)$ and $L(e)$ denote the clusters and categories related to e. The multiplicity precision $Mult.Pre\,(e, e')$ is 1 when e and e' in the same cluster share the same category. Therefore, the B-Cubed precision of one item is its averaged multiplicity precision with the other items in the same categories. The multiplicity recall $Mult.Recall\,(e, e')$ is 1 when e and e' in the same category share the same cluster. Similarly, the harmonic mean of B-Cubed precision and recall is defined by,

$$F = \frac{1}{\alpha \frac{1}{BEP} + (1-\alpha)\frac{1}{BER}} \quad \alpha = \{0.2, 0.5\}$$

5.3 Document Clustering for WPD

The clustering algorithm used in this work is the hierarchical agglomerative clustering algorithm in single linkage (Manning et al., 2008). Documents are represented by key phrase vectors and their similarities are computed using the cosine metric. The weight for a key phrase is calculated with the consideration of both TF and ITF as well as the link probability as dedefined before (similar to that used in (Xu et al., 2012).

$$W_k = \log(TF(k) + 1) * (\log IDF(k) + Pr_{link}(k))$$

where $TF(k)$ denotes the term frequency of k, $IDF(k)$ is the inverse document frequency of k and $Pr_{link}(k)$ is the link probability of k. $Pr_{link}(k)$ is defined as $Pr_{link}(k) = \frac{C_{link}(k)}{C_{occur}(k)}$. $C_{link}(k)$ is the number of hyperlinks anchored to k in Wikipedia, and $C_{occur}(k)$ is the number of occurrences of k in the Wikipedia articles. That means some extracted key phrases appear in the Wikipedia articles, but are not linked to, thus their importance decreases.

As the number of clusters cannot be predetermined, we use the $WePS1$ development data to select optimal parameters which give the best B-Cubed and purity F-measures. The parameter configurations are listed in Table 2.

SD	$DF(\lambda)$	WS	CP
3	0.5	20	0.182

Table 2: Parameter Configurations

SD denotes the skip distance which is used to specify how many gaps can be allowed in a skip bigram; DF refers to the decay factor λ which is used to penalize the non-continuous skip bigrams. WS is the window size to specify the maximum number of key phrases surround a name mention, and CP denotes the cut-off point for the number of clusters in the hierarchical dendrogram.

In the following experiments, A_{PKPB} refers to the clustering algorithm purely using key phrases ($PKPB$ stands for pure key phrase

based approach) and A_{SKIP} denote the clustering algorithm using skip bigrams. The aggregated algorithm is denoted by A_{AKPC}. Table 3 and Table 4 show the comparison of A_{AKPC} the algorithm with the top-3 systems in $WePS$ 2009 in terms of purity measure and B-Cubed measure, respectively.

Runs	F-measures			
	α=0.5	α=0.2	Pur.	Inv_Pur.
T1: PolyUHK	**0.88**	**0.87**	0.91	0.86
T2: UVA_1	0.87	0.87	0.89	0.87
T3: ITC_UT_1	0.87	0.83	**0.95**	0.81
A_{AKPC}	**0.88**	**0.87**	0.86	**0.87**

Table 3: Performance Comparison of A_{AKPC} using Purity scores

Runs	F-measures			
	α=0.5	α=0.2	BEP	BER
T1:PolyUHK	**0.82**	**0.80**	0.87	0.79
T2:UVA_1	0.81	0.80	0.85	**0.80**
T3:ITC_UT_1	0.81	0.76	**0.93**	0.73
A_{AKPC}	**0.82**	**0.80**	0.85	**0.80**

Table 4: Performance Comparison of A_{AKPC} using B-Cubed scores

Table 3 and Table 4 show that in comparison to the top 1 system, the proposed A_{AKPC} has the same performance in terms of F-measure for both purity score and B-cubed score. In terms of B-Cubed recall, A_{AKPC} achieves the highest score, implying that the number of categories has been well guaranteed by our clustering solutions. Admittedly, our system loses 2 percent in terms of B-Cubed precision. However, when comparing to the features used in the top 3 systems, the top 1 system by $PolyUHK$ (Chen et al., 2009) incorporates tokens, title tokens, n-gram and snippet features into its system using VSM. The $PolyUHK$ system has to tune the unigram and bigram weights through the Goodgle 1T corpus which is external to the WePS training data. The second best UVA_1 system (Balog et al., 2009) employs all tokens of in the training document only documents, and the third best ITC_UT_1 system (Ikeda et al., 2009) uses named entities, compound nouns and URL links features. The A_{AKPC} algorithm in this paper simply uses key phrase and limited amount of skip bigrams around the name mentions. The Key phrase extraction algorithm are trained by Wikipedia article. Even though this takes additional computation power, it can be done once only. In the testing phase, extraction of key phrases is much faster than the other systems and the dimension of the key phrases in the VSM is also much smaller than the other systems.

To measure the effectiveness of the two sub-algorithms A_{PKPB} and A_{SKIP}, performance of the two algorithms are also evaluated separately as independent clustering algorithms shown in Table 5 and Table 6 for B-cubed measures and purity measures, respectively. Note that when evaluating the two algorithms, the cut-off points need to be readjusted from the $WEPS1$ development data. The cut-off point for A_{PKPB} remains unchanged as 0.182 and the A_{SKIP} cut-off point is set to 0.055.

From Table 5 and Table 6, it can be seen that for both A_{PKPB} and A_{SKIP}, if used separately, do not perform as well as A_{AKPC}. However, A_{PKPB} has a better performance than A_{SKIP} when used alone. This implies that key phrases, as a single feature in clustering algorithm, are better features than using skip bigrams of key phrases surrounding the mentions. This is easy to understand as the context windows for the name mentions used in skip bigram model do not have as large a coverage of key phrases as that in the whole documents. However, A_{SKIP} gives the second best performance in B-Cubed precision and purity. This is why the overall B-Cubed precision and purity are improved after its aggregation.

In terms of purity in Table 5, A_{PKPB} has the same performance as the top 1 and top 2 systems in term F0.2 and 1 percent better when compared to the top 2 system in terms of inverse purity. Our system, however, loses 1 percent in F0.5 score and 4 percent in purity score when compared to the top 1 system and A_{SKIP} achieves a second best purity score among the top 3 systems.

It is most important to point out that the A_{PKPB} algorithm in Table 6, however simple, has a competitive performance in comparison to the top 1 system. A_{PKPB} has the best results in terms of F0.2 for B-cubed score, implying that

Runs	F-measures			
	α=0.5	α=0.2	Pur.	Inv_Pur.
T1: PolyUHK	**0.88**	**0.87**	0.91	0.86
T2: UVA_1	0.87	0.87	0.89	0.87
T3: ITC_UT_1	0.87	0.83	**0.95**	0.81
A_{PKPB}	0.87	**0.87**	0.87	**0.88**
A_{SKIP}	0.79	0.74	**0.93**	0.71

Table 5: Performance Comparison of A_{PKPB} and A_{SKIP} using Purity scores

two documents in the same manually annotated categories share the same cluster produced by our system. In terms of B-Cubed score, even though A_{PKPB} loses one percent in F0.5, the performance gain is three percent in B-Cubed recall when compared to the $PolyUHK$ system. In terms of B-Cubed precision, our system is not as good as the top three systems. However, our system strikes a better balance between B-Cubed precision and purity score, which means that our system's clustering solutions are consistent with manually annotated categories.

Runs	F-measures			
	α=0.5	α=0.2	BEP	BER
T1:PolyUHK	**0.82**	**0.80**	0.87	0.79
T2:UVA_1	0.81	0.80	0.85	**0.80**
T3:ITC_UT_1	0.81	0.76	**0.93**	0.73
A_{PKPB}	0.81	**0.81**	0.82	**0.82**
A_{SKIP}	0.69	0.63	**0.91**	0.60

Table 6: Performance Comparison of A_{PKPB} and A_{SKIP} using B-Cubed scores

In order to demonstrate the performance improvement by aggregating the skip bigrams into the vector space model, we looked at our designs with and without aggregating skip bigrams. Table 7 shows the evaluation results.

Runs	F-measures		Purity		B-Cubed	
	B-Cubed	Purity	P	IP	BEP	BER
A_{PKPB}	0.81	0.87	0.87	**0.88**	0.82	**0.82**
A_{AKPC}	**0.82**	**0.88**	**0.89**	0.87	**0.85**	0.80

Table 7: Performance Comparison of A_{PKPB} and A_{AKPC} using both Purity and B-Cubed scores

In Table 7, both B-Cubed and purity F0.5 scores have been increased by 1 percent. The B-Cubed precision is improved by 3% and purity is increased by 2%, which means that the A_{AKPC} gives a much more reliable clustering solution. It is common in most information retrieval cases that algorithms with high precision will have a compromise on their recall performance. In this paper, we have gained 3% and 2% improvement in B-Cubed precision and Purity, but lost 2% and 1% in B-Cubed recall and inverse purity, respectively.

6 Conclusions and Future Work

This paper proposed the $AKPC$ algorithm to use key phrases as document representations and skip-bigram of key phrases as contextual information in Web person disambiguation. Results show that the proposed $AKPC$ algorithm gives a competitive performance when compared to the top three systems in $WePS$ 2009.

Further investigation also shows that clustering based on key phrases as single features is very effective. It employs a supervised approach to extract meaningful key phrases for person names. The extraction of key phrases in the training phase is fully automatic and no manual annotation is needed as the training data is from Wikipedias anchor text. The weighting scheme takes into consideration of both the traditional TF*IDF and the Wikipedia link probability. Experiments show that the proposed key phrase based clustering algorithm using VSM is both effective and efficient. Unlike the tokens used by most of previous researches, key phrases are more meaningful and are more capable of separating people of the same namesake.

Further extension of this work includes aggregating order-sensitive skip bigrams into key phrase-based vector space model to enrich context information in the inclusion of web persons disambiguation. Experiments show that the precision of clustering solutions is increased. We combined the decay factor with the skip distance to assign a reasonable weight for skip bigrams and studied the effectiveness of varying skip distance and decaying factor. In future work, we will explore skip ngrams for a larger n. Moreover, we will explore the use of efficient combination of key phrases with skip ngrams.

References

A. Hulth. 2003. Improved Automatic Keyword Extraction Given more Linguistic Knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216-223.

A. Hulth and B. Megyesi. 2006. A Study on Automatically Extracted Keywords in Text Categorization. In *CoLing/ACL 2006, Sydney*.

Y. Chen, Sophia Yat Mei Lee and Chu-Ren Huang. 2009. PolyUHK: A Robust Information Extraction System for Web Personal Names. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*.

Chong Long and Lei Shi. 2010. Web Person Name Disambiguation by Relevance Weighting of Extended Feature Sets. In *Third Web People Search Evaluation Forum (WePS-3), CLEF 2010*.

Decong Li, Sujian Li, Wenjie Li, Wei Wang, and Weiguang Qu. 2010. A Semi-Supervised Key Phrase Extraction Approach: Learning from Title Phrases through a Document Semantic Network. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 296-300.

D.C. Manning, P. Raghavan, and H. Schutze. 2008. *Hierarchical Clustering. Introduction to Information Retrieval*. Cambridge University Press, New York, 2008, 377-401.

E. Frank, G. W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific Keyphrase Extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI99)*, pages 668-673.

G. Salton and M. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.

Hongyuan Zha. 2002. Generic Summarization and Key Phrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Tampere*, pages 113-120.

I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin and C.G. Nevill-Manning. 2000. KEA: Practical Automatic Keyphrase Extraction. In *Working Paper 00/5, Department of Computer Science, The University of Waikato*.

J. Artiles, J. Gonzalo and S. Sekine. 2007. The SemEval-2007 WePS Evaluation: Establishing a Benchmark for the Web People Search Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*,pages 64–69.

J. Artiles, J. Gonzalo and S. Sekine. 2009. Weps 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*.

J. Artiles, A. Borthwick, J. Gonzalo, S. Sekine and E. Amigo. 2010. WePS-3 Evaluation Campaign: Overview of the Web People Search Clustering and Attribute Extraction Tasks. *Third Web People Search Evaluation Forum (WePS-3), CLEF 2010*.

Jian Xu, Qin Lu, and Zhengzhong Liu. 2012. PolyUCOMP: Combining Semantic Vectors with Skip bigrams for Semantic Textual Similarity. *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (*SEM 2012)*.

Jian Xu, Qin Lu and Zhengzhong Liu. 2012. Combining Classification with Clustering for Web Person Disambiguation. *WWW 2012 Companion, April 16-20, 2012, Lyon, France*.

K. M. Hammouda, D.N. Matute and M.S. Kamel. 2005. CorePhrase: Keyphrase Extraction for Document Clustering. In *Proceedings of MLDM. 2005*.

K. Balog, J. He, K. Hofmann, et al. 2009. The University of Amsterdam at WePS2. *2nd Web People Search Evaluation Workshop. 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*.

Lin Chin-Yew and Franz Josef Och. 2004a. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*.

M. Ikeda, S. Ono, I. Sato, M. Yoshida, and H. Nakagawa. 2009. Person Name Disambiguation on the Web by Two-Stage Clustering. *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*.

Marina Litvak and Mark Last. 2008. Graph-based Keyword Extraction for Single-document Summarization. *In Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization,* pages 17-24.

M. Steyvers and T. Griffiths. 2007. Probabilistic Topic Models. *T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds), Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum.*

O. Popescu and B. Magnini. 2007. IRST-BP: Web People Search using Name Entities. *Proceedings of the Fourth International Workshop on Semantic*

Evaluations (SemEval-2007), June (2007), pages 195-198.

Xiaojun Wan and Jianguo Xiao. 2008. CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969-976.

Xianpei Han and Jun Zhao. 2009. ASIANED: Web Personal Name Disambiguation Based on Professional Categorization. *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, pages 2-5.

Zhiyuan Liu,Wenyi Huang, Yabin Zheng and Maosong Sun. 2010. Automatic Keyphrase Extraction via Topic Decomposition. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 366-376.