ANANYSIS OF MODIFIED INVERSE DOCUMENT FREQUENCY VARIANTS FOR WORD SENSE DISAMBIGUATION

[1]ZUN MAY MYINT, [2]MAY ZIN OO

[1,2]Department of Computer Engineering and Information Technology,
Mandalay Technological University, Myanmar
E-mail: [1]zunmaymyint.zmm@gmail.com, [2]mayzinoo.mz@gmail.com

Abstract— Word Sense Disambiguation (WSD) is a method to find the correct sense of ambiguous word from the existing senses by calculating the similarity between ambiguous words in Information Retrieval system. In WSD process, there are many WSD approaches, in which K-Nearest Neighbour (KNN) is used because it is extremely simple and effective in text classification. The cosine similarity method in KNN, in which term frequency and inverse document frequency (TF-IDF) scheme is used to calculate the weight of each word. There is a challenge that the original TF-IDF scheme eliminates the related senses in WSD process although there is a related sense. This paper thus proposes the three modified TF-MIDF methods to solve the no-relevant problem by modifying the IDF equation and analyses the modified IDF methods to ascertain which MIDF method can improve the performance of WSD method.

Keywords— WSD, KNN Classifier, Cosine Similarity, Modified Inverse Document Frequency (MIDF), WordNet.

I. INTRODUCTION

The goal of Information Retrieval (IR) is to provide users with documents that will satisfy their information need. Ambiguities in IR system is a big central challenge to retrieve a correct sense of searched terms. IR system will improve its performance if the query it retrieves are represented by word senses rather than words.

Words have multiple meaning, and WSD is a task to determine the proper meaning of word. There are many approaches in WSD. A rich variety of techniques have been researched from dictionary-based methods that use knowledge encoded in lexical resources, supervised machine learning works on classifiers, semi-supervised learning method and unsupervised learning method supports clusters. Word sense disambiguation is essential for many applications such as information retrieval, information extraction, text summarization, and all tasks in a text mining framework. To support these applications, this system proposes the supervised learning WSD method. This method is based on KNN classifier and WordNet. WordNet is also used as the lexical resource to extract hypernyms, hyponyms, synonyms and gloss of ambiguous word. In this sense, the Cosine similarity method is used to choose the correct sense that is relevant to the ambiguous word in KNN by calculating the similarity. The Term Frequency and Inverse Document Frequency (TF-IDF) scheme in cosine similarity is used for calculating the weight of each word and the original IDF eliminates the related senses. So, this paper proposes the modified TF-IDF methods that are created with different point of view to solve no-relevant problem.

The rest of the paper is organized as follows: the proposed problem in original TF-IDF is described together with an example in section 2. Section 3 proposes the modified TF-IDF methods. Section 4 analyses the similarity results between original and modified TF-IDF. Finally, conclusion is given.

II. PROPOSED PROBLEM

The similarity between the testing vector and training vectors in KNN classifier is commonly based on the similarity function. There are many similarity or distance functions. After converting each context to a vector of words, this system uses the cosine similarity method to measure the similarity between a new context and each existing context in the training corpus. The cosine similarity between training vector d_j and testing vector q can be calculated as the following equation. Cosine similarity between two typical vectors can be defined as follow:

$$\text{cosine}(d_j, q) = \frac{\sum_{i=1}^{|v|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|v|} (w_{ij})^2} \times \sqrt{\sum_{i=1}^{|v|} (w_{iq})^2}} \quad (1)$$

where, cosine(d_j, q) is cosine similarity between training vector d_j and testing vector q. W_{ij} is weight of the term t_i within training vector d_j. W_{iq} is weight of the term t_i within testing vector q [5].

A. Original Term Frequency and Inverse Document Frequency (TF-IDF) Weighting Scheme

To calculate the weight of each term in the training vector and testing vector, the TF- IDF weighting scheme is used. The term frequency is multiplied with the inverse document frequency to obtain the weight of each term.

The term frequency within training vector is as follows:

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|v|j}\}} \quad (2)$$

where, f_{ij} is the raw frequency count of term t_i in training vector d_j and tf_{ij} is the normalize term frequency of term t_i in training vector d_j. The inverse document frequency (IDF) is as follow:

$$idf_i = \log \frac{N}{df_i} \quad (3)$$

where, df_i is the number of train vectors in which term t_i appear. N is the total number of train vectors in the system. idf_i is the inverse document frequency of term t_i.

The weight of the term within the training vector is as follow:

$$w_{ij} = tf_{ij} \times idf_i \quad (4)$$

where, w_{ij} is the weight of the term t_i in training vector dj.

The weight of the term within testing vector is as follows:

$$w_{iq} = \left[0.5 + \frac{0.5 f_{iq}}{\max\{f_{1q}, f_{2q}, ..., f_{|v|q}\}} \right] \times \log\left(\frac{N}{df_i} \right) \quad (5)$$

where, w_{iq} is the weight of the term t_i in testing vector q and f_{iq} is the raw frequency count of term t_i in the testing vector q [5].

B. Explanation of the System

As an example, the input query is as follows:

Input Query: Classification of training samples into each category

After receiving the input query, stopwords are removed and keywords are extracted from this query. In this query, "classification", "training", "samples" and "category" are keywords that are meaningful or important words. For ambiguous word and disambiguous word classification, the gloss of synonyms, hypernyms and hyponyms of each sense from the WordNet are used.

In this sample, "classification", "training" "samples" and "category" are ambiguous words because they have each sense from the WordNet. Sample word senses and its gloss from WordNet is shown in Table 1.

Table 1: Sample Word Senses and Its Gloss from WordNet

ID	Keyword	Sense ID	Sense Name	Gloss
1	classification	Sense 1	categorization, assortment	the act of assigning someone or something to a particular class or category.
		Sense2	categorization	a group of people or things arranged by class or category.
		Sense3	categorization, sorting	incorporating something under a general category.
2	samples	Sense1	distribution	items selected at random from a population and used to test hypotheses about the population.
3	category	Sense1	class	a collection of things sharing a common attitude; there are two classes of detergents.

By using testing vector and each training vector, this system searches the most relevant sense of each ambiguous word by using KNN classifier, cosine similarity and weighting scheme. In this sample, this system defines "K = 1" so that it chooses the most relevant one sense that has the highest similarity value among other similarity results. The training vector of each sense is created by using the gloss of word sense. For the training vector, the stopwords are removed from the gloss. As a sample to create training and testing vectors,

For "classification" keyword;

Training vector 1 for sense 1 from WordNet: [act, assigning, someone, something, particular, class, category]

Training vector 2 for sense 2 from WordNet: [group, people, things, arranged, class, category]

Training vector 3 for sense 3 from WordNet: [incorporating, something, under, general, category]

Testing vector from input query: [training, samples, category]

By using the original TF and IDF weighting scheme, the weight results for the "classification" keyword are shown in Table 2.

Table 2: Weight Result Using TF -IDF about "Classification" Keyword

Vector Name	Term (Keyword)	Weight Result from TF-IDF		
		TF-IDF		Weight
		TF	IDF	
Training Vector 1	act	1	0.47712	0.47712
	assigning	1	0.47712	0.47712
	someone	1	0.47712	0.47712
	something	1	0.47712	0.47712
	particular	1	0.47712	0.47712
	class	1	0.47712	0.47712
	category	1	0	0
Training Vector 2	group	1	0.47712	0.47712
	people	1	0.47712	0.47712
	things	1	0.47712	0.47712
	arranged	1	0.47712	0.47712
	class	1	0.47712	0.47712
	category	1	0	0
Training Vector 3	incorporating	1	0.47712	0.47712
	something	1	0.47712	0.47712
	under	1	0.47712	0.47712
	general	1	0.47712	0.47712
	category	1	0	0
Testing Vector	training	1	0.47712	0.47712
	samples	1	0.47712	0.47712
	category	1	0.47712	0.47712

For "category" keyword;

Training vector for sense 1 from WordNet: [collection, things, sharing, common, attribute, classes, detergents]

Testing vector from input query: [classification, training, samples]

Table 3: Weight Result Using TF -IDF About "Category" Keyword

Vector Name	Term (Keyword)	Weight Result from TF-IDF		
		TF-IDF		Weight
		TF	IDF	
Training Vector	collection	1	0	0
	things	1	0	0
	sharing	1	0	0
	common	1	0	0
	attribute	1	0	0
	classes	1	0	0
	detergents	1	0	0
Testing Vector	classification	1	0	0
	training	1	0	0
	samples	1	0	0

By using the original TF and IDF weighting scheme, the weight results for "category" keyword are shown in Table 3.

This system calculates the similarity between the training vector and the testing vector by using the weight result of the original TF-IDF. According to the similarity results for the "classification" keyword, there is no sense although there is relevant to the user query. For the "category" and "samples" keyword, there is no relevant sense according to the similarity result. Therefore, the system proposed the modified IDF methods to improve the performance of the WSD process.

III. PROPOSED SOLUTION

The proposed system presents three modified inverse document frequency (MIDF) methods to support the performance and correctness of similarity method.

A. Proposed Modified Inverse Document Frequency

In the following methods, dfi is the number of train vectors in which term ti appear. N is the total number of train vectors in the system. The midfi is the modified inverse document frequency of term ti.

1) Modified-1 Inverse Document Frequency (M-1 IDF):M-1 IDF method is as follows:

$$midf_i = \log\left(\frac{N}{df_i} + \frac{N}{df_i}\right) \qquad (6)$$

2) Modified-2 Inverse Document Frequency (M-2 IDF):M-2 IDF method is as follows:

$$midf_i = \log\left(\frac{N}{df_i} + \frac{N}{df_i} + \frac{N}{df_i}\right) \qquad (7)$$

3) Modified-3 Inverse Document Frequency (M-3 IDF):M-3 IDF method is as follows:

$$midf_i = \log\left(\frac{N}{df_i} + \frac{N}{df_i} + \frac{N}{df_i} + \frac{N}{df_i}\right) \qquad (8)$$

B. Explanation of the Proposed System

This system searches the most relevant sense of each ambiguous word by using the three TF-MIDF weighting schemes in the KNN classifier. The same sample query is also used in the explanation of the TF-MIDF scheme.

Input Query: Classification of training samples into each category

1) TF-M1 IDF Weighting Scheme: By using the term frequency (TF) and modified-1 IDF (M1IDF) weighting scheme, the weight results of the "category" keyword are shown in Table 4.

Table 4: Weight Result Using TF- M1 IDF about "Category" Keyword

Vector Name	Term (Keyword)	Weight Result from TF- M1 IDF		
		TF-M1IDF		Weight
		TF	M1 IDF	
Training Vector	collection	1	0.30102	0.30102
	things	1	0.30102	0.30102
	sharing	1	0.30102	0.30102
	common	1	0.30102	0.30102
	attribute	1	0.30102	0.30102
	classes	1	0.30102	0.30102
	detergents	1	0.30102	0.30102
Testing Vector	classification	1	0.30102	0.30102
	training	1	0.30102	0.30102
	samples	1	0.30102	0.30102

For "classification" keyword, calculated weight results using TF-M1 IDF are shown in Table 5. By using the weight result, cosine similarity method is used to calculate the similarity between the training vector and testing vector of the "classification" keyword and "category" keyword.

Table 5: Weight Result Using TF –M1 IDF about "Classification" Keyword

Vector Name	Term (Keyword)	Weight Result from TF- M1 IDF		
		TF M1 IDF		Weight
		TF	M1 IDF	
Training Vector 1	act	1	0.77815	0.77815
	assigning	1	0.77815	0.77815
	someone	1	0.77815	0.77815
	something	1	0.77815	0.77815
	particular	1	0.77815	0.77815
	class	1	0.77815	0.77815
	category	1	0.30102	0.30102
Training Vector 2	group	1	0.77815	0.77815
	people	1	0.77815	0.77815
	things	1	0.77815	0.77815
	arranged	1	0.77815	0.77815
	class	1	0.77815	0.77815
	Category	1	0.30102	0.30102
Training Vector 3	incorporating	1	0.77815	0.77815
	something	1	0.77815	0.77815
	under	1	0.77815	0.77815
	general	1	0.77815	0.77815
	category	1	0.30102	0.30102
Testing Vector	training	1	0.77815	0.77815
	samples	1	0.77815	0.77815
	category	1	0.77815	0.77815

According to the similarity result using TF-M1IDF, the sense "categorization sorting" is relevant to the "classification" keyword.

2) TF-M2 IDF Weighting Scheme: By using the TF and modified-2 IDF (M2 IDF) weighting scheme, the weight results are shown in Table 6.

Table 6: Weight Result Using TF–M2 IDF about "Category" Keyword

Vector Name	Term (Keyword)	Weight Result from TF- M2 IDF		
		TF-M2IDF		Weight
		TF	M2 IDF	
Training Vector	collection	1	0.47712	0.47712
	things	1	0.47712	0.47712
	sharing	1	0.47712	0.47712
	common	1	0.47712	0.47712
	attribute	1	0.47712	0.47712
	classes	1	0.47712	0.47712
	detergents	1	0.47712	0.47712
Testing Vector	classification	1	0.47712	0.47712
	training	1	0.47712	0.47712
	samples	1	0.47712	0.47712

For "category" and "classification" keywords, calculated weight results using TF-M2 IDF are shown in Table 6 and 7. According to the weight results by using TF-M2 IDF, the similarity between the training vector and testing vector can be calculated by using cosine similarity method.

Table 7: Weight Result Using TF –M2 IDF About "Classification" Keyword

Vector Name	Term (Keyword)	Weight Result from TF- M2 IDF		
		TF- M2 IDF		Weight
		TF	M2 IDF	
Training Vector 1	act	1	0.95424	0.77815
	assigning	1	0.95424	0.77815
	someone	1	0.95424	0.77815
	something	1	0.95424	0.77815
	particular	1	0.95424	0.77815
	class	1	0.95424	0.77815
	category	1	0.47712	0.47712
Training Vector 2	group	1	0.95424	0.77815
	people	1	0.95424	0.77815
	things	1	0.95424	0.77815
	arranged	1	0.95424	0.77815
	class	1	0.95424	0.77815
	category	1	0.47712	0.47712
Training Vector 3	incorporating	1	0.95424	0.77815
	something	1	0.95424	0.77815
	under	1	0.95424	0.77815
	general	1	0.95424	0.77815
	category	1	0.47712	0.47712
Testing Vector	training	1	0.95424	0.95424
	samples	1	0.95424	0.95424
	category	1	0.95424	0.95424

According to the similarity result using TF-M2 IDF, the sense "categorization sorting" is relevant to the "classification" keyword.

3) TF-M3 IDF Weighting Scheme: By using the TF and modified-3 IDF (M3 IDF) weighting scheme, the weight results of the "category" keyword are shown in Table 8.

Table 8: Weight Result Using TF -M3 IDF about "Category" Keyword

Vector Name	Term (Keyword)	Weight Result from TF M3 IDF		
		TF-M3 IDF		Weight
		TF	M3 IDF	
Training Vector	collection	1	0.60205	0.60205
	things	1	0.60205	0.60205
	sharing	1	0.60205	0.60205
	common	1	0.60205	0.60205
	attribute	1	0.60205	0.60205
	classes	1	0.60205	0.60205
	detergents	1	0.60205	0.60205
Testing Vector	classification	1	0.60205	0.60205
	training	1	0.60205	0.60205
	samples	1	0.60205	0.60205

Table 9: Weight Result Using TF –M3 IDF about "Classification" Keyword

Vector Name	Term (Keyword)	Weight Result from TF- M3 IDF		
		TF- M3 IDF		Weight
		TF	M3 IDF	
Training Vector 1	act	1	1.07918	1.07918
	assigning	1	1.07918	1.07918
	someone	1	1.07918	1.07918
	something	1	1.07918	1.07918
	particular	1	1.07918	1.07918
	class	1	1.07918	1.07918
	category	1	0.60205	0.60205
Training Vector 2	group	1	1.07918	1.07918
	people	1	1.07918	1.07918
	things	1	1.07918	1.07918
	arranged	1	1.07918	1.07918
	class	1	1.07918	1.07918
	category	1	0.60205	0.60205
Training Vector 3	incorporating	1	1.07918	1.07918
	something	1	1.07918	1.07918
	under	1	1.07918	1.07918
	general	1	1.07918	1.07918
Testing Vector	category	1	0.60205	0.60205
	training	1	1.07918	1.07918
	samples	1	1.07918	1.07918
	category	1	1.07918	1.07918

For "category" and "classification" keywords, calculated weight results using TF-M3 IDF are shown in Table 8 and 9. By using TF-M3 IDF, the sense "categorization sorting" is relevant to the "classification" keyword.

IV. ANALYSIS OF SIMILARITY RESULT

For disambiguation, this system uses weight results and also calculates the similarity between training vector and testing vector to choose the most relevant sense of ambiguous word. According to the "K = 1", this system chooses the sense that has the highest

similarity value. The similarity results about "classification" keyword are shown in Table 10. According to the different weighting scheme, the TF-M1 IDF, TF- M2 IDF and TF-M3 IDF can provide the better performance of WSD method because the modified IDF weighting schemes point out the similarity between training vector and testing vector.

Table 10: Similarity Result about "Classification" Keyword

ID	Weighting Scheme	Similarity Result between Training Vector 3 and Testing Vector
1	Original TF-IDF	0
2	TF- M1 IDF	0.10964
3	TF- M2 IDF	0.14003
4	TF- M3 IDF	0.15512

The original TF-IDF cannot retrieve relevant sense about ambiguous word because the weighting scheme of TF-IDF cannot produce similarity result between training vector and testing vector.

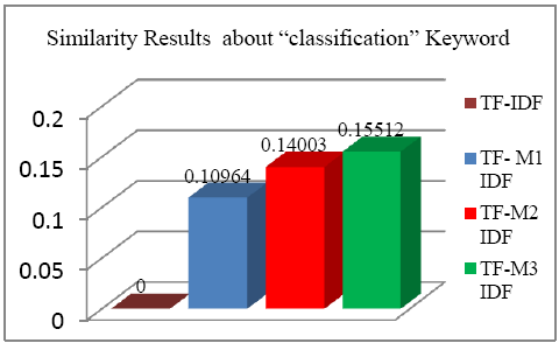

Fig. Similarity Result about "classification" keyword

Therefore, the original TF-IDF produces the output disambiguated user query as the input query although there is relevant sense to the query. On the other hand, the modified schemes, such as TF- M1 IDF, TF-M2 IDF and TF-M3 IDF, can give the similarity values and retrieve the disambiguated output query. When the modified schemes are tested by increasing one or more factors, the results verified that more factors give higher similarity values. However, it cannot be assumed as better performance because the increasing factors of IDF give same disambiguated output query.

To know which weighting scheme is more effective for WSD process by increasing one or more factors of IDF equation, all modifications give the same relevant sense "categorization sorting" for the ambiguous word "classification". Therefore, when the computation complexity and the processing time are considered, the variation of increasing factors that simply add N/df_i once to the original IDF equation is more effective for WSD process.

Output Disambiguated User Query: Classification categorization sorting of training samples into each category

CONCLUSION

Word Sense Disambiguation process will be better if ambiguous words can be correctly disambiguated. This paper proposes the modified IDF schemes of cosine similarity in KNN classifier for the performance improvement of WSD process The original TF-IDF can retrieve no relevant sense about ambiguous word because this weighting schemes cannot search the similarity between training vector and testing vector. The proposed MIDF schemes which provide the better performance by simply adding N/dfi to avoid the problem of no-relevant sense retrievement of IDF method. By comparing the performance between KNN classifier based on the original IDF scheme and KNN classifier based on different MIDF schemes, the variation of factor which simply add once to the original IDF scheme which solve no relevant problem is more effective for WSD system.

REFERENCES

[1] Singh. S and Siddiqui. T "Utilizing Corpus Statistics for Hindi Sense Disambiguation", the International Arab Journal of Information Technology, Vol. 12, No. 6A, Department of Electronics and Communication, University of Allahabad, India, 2015.

[2] Nasir. J. A and Karim. A "A knowledge-based Semantic Kernel for Text Classification", Biotechnology Center, Technische University at Dresden , Germany, 2010.

[3] Donald Metzler "Generalized Inverse Document Frequency", Napa Valley, California, USA, October 26-30, 2008.

[4] Martin Wanton. T"A clustering –based Approach for Unsupervised Word Sense Disambiguation", Department of Language and Computer Systems, Universidad de Educacion a Distancia (UNED), Madrid, 2012.

[5] B. Liu, "Web Data Mining", Department of Computer Science, University of Illinois at Chicago, USA, Springer-Verlag Berlin Heidelberg, 2007.

★★★