Concept-Based Mining Model for Enhancing Text Clustering

Mrs.D.M.Kulkarni [1], [2] Mr.S.K.Shirgave

[1] *IT Department Dkte's TEI Ichalkaranji (Maharashtra), India*

[2]*IT Department Dkte's TEI Ichalkaranji (Maharashtra) ,India*

Kulkarni.dhanashri@gmail.com , skshirgave@yahoo.com

ABSTRACT

Text mining techniques are based on the statistical analysis of a term, either word or phrase. Statistical analysis of a term frequency captures the importance of the term within a document only. However, two terms can have the same frequency in their documents, but one term contributes more to the meaning of its sentences than the other term. Thus, this text mining model should indicate terms that capture the semantics of text. In this case, the mining model can capture terms that present the concepts of the sentence, which leads to discovery of the topic of the document. A new concept-based mining model that analyzes terms on the sentence, document, and corpus levels is introduced. This model can effectively discriminate between non important terms with respect to sentence semantics and terms which hold the concepts that represent the sentence meaning. The proposed model consists of sentence, document, corpus based concept analysis, and concept-based similarity measure. The term which contributes to the sentence semantics is analyzed on the sentence, document, and corpus levels rather than the traditional analysis of the document only. The proposed model can efficiently find significant matching concepts between documents, according to the semantics of their sentences.

Keywords—Concept-based mining model, sentence-based, document-based, corpus-based, concept analysis, conceptual term frequency, concept-based similarity.

INTRODUCTION

Document clustering is an optimization process which attempts to determine a partition of the document collection so that documents within the same cluster are as similar as possible and the discovered clusters as separate as possible. Document clustering algorithms are used in a variety of tasks and applications for facilitating automatic organization, browsing, Summarization, and retrieval of structured and unstructured documents.

Most of the clustering techniques use TF_IDF method. But this method has some drawbacks:

These techniques generally fail to differentiate the degree of semantic importance of each term, and assign weights without distinguishing between semantically important and unimportant words within the document. They do not consider synonyms, polysemous etc. To overcome the lack of semantic consideration we are moving to concept based text clustering. In this the mining model can capture terms that present the concepts of the sentence, which leads to discovery of the topic of the document. A new concept-based mining model that analyzes terms on the sentence, document, and corpus levels is introduced. The concept-based mining model can effectively discriminate between non important terms with respect to sentence semantics and terms which hold the concepts that represent the sentence meaning. The proposed mining model consists of sentence-based concept analysis, document-based concept analysis, corpus-based concept-analysis, and concept-based similarity measure. The term which contributes to the sentence semantics is analyzed on the sentence, document, and corpus levels rather than the traditional analysis of the document only. In this can efficiently find significant matching concepts between documents, according to the semantics of their sentences. The similarity between documents is calculated based on a new concept-based similarity measure. The proposed similarity measure takes full advantage of using the concept analysis measures on the sentence, document, and corpus levels in calculating the similarity between documents.

How to extract semantic structure of a sentence The semantic structure of a sentence can be characterized by a form of **verb argument structure**. This underlying structure allows the creation of a composite meaning representation from the meanings of the individual concepts in a sentence. The verb argument structure permits a link between the arguments in the surface structures of the input text and their associated semantic roles. Before finding the verb argument structure of a sentence we need to label the parts of speech for words present in those sentences.

VERB ARGUMENT STRUCTURE

In linguistics, a verb argument is a phrase that appears in a syntactic relationship with the verb in a clause. In English, the two most important arguments are the subject which is the noun phrase that appears before the verb and the direct object, which normally appear after the verb. These are called **core arguments.** In syntax, a subject within linguistics, the **sub categorization frame** of a word is defined to be the number and types of syntactic arguments that it co-occurs with. The following examples present some common sub-categorization frames in English language:

Intransitive Verb: NP [subject]: It accepts only the subject argument.

Ex:"The man walked"

Transitive Verb: NP [subject], NP [object]: Accept an optional object argument

Ex:"Andrew hit the ball".

Ditransitive Verb: NP [subject], NP [object], NP [indirect object]: Accepts a third indirect argument also.

Ex:"Mike sent John an email".

Similarly, there are semantic regularities which are called selectional restrictions or selectional preferences. Selectional restriction is a semantic constraint forced by a lexeme on the concepts that can fill the various argument roles associated with it. For example: the verb bark prefers dogs as subject. The verb eat prefers edible things as objects.

LITERATURE SURVEY

Most current document clustering methods are based on the Vector Space Model (VSM) [3],[4]which is a widely used data representation for text classification and clustering. The VSM represents each document as a feature vector of the terms (words or phrases) in the document. Each feature vector contains term weights (usually term frequencies)of the terms in the document. The similarity between the documents is measured by one of several similarity measures that are based on such a feature vector. Examples include the cosine measure and the Jaccard measure[2]. Methods used for text clustering include decision trees conceptual clustering clustering based on data summarization, statistical analysis , neural nets inductive logic programming , and rule-based systems among others. In text clustering, it is important to note that selecting important features, which present the text data properly, has a critical effect on the output of the clustering algorithm. Moreover, weighting these features accurately also affects the result of the clustering algorithm substantially.

VECTOR SPACE MODEL

Before we represent documents as tf-idf vectors, we need some preprocessing. **There are commonly two steps:**

First, Remove stop words, such as 'a', 'any', 'what', 'I', etc, since they are frequent and carry no information. A stop words list can be found online.

Second, stem the word to its origin, which means we only consider the root form of words. For example, ran, running, runs are all stemmed to run, and happy, happiness, happily are all stemmed to happi. There are certain criteria, and the standard algorithm is the Porter's stemmer, which is also free online. A more elaborate way of stemming is by using the m WordNet, which in addition to su_x-stripping also groups words into synsets, and leads to an ontology-based (instead of word-based) document clustering method. Vector Space Model is the basic model for document clustering, upon which many modi_ed models are based. We briey review a few essential topics to provide a su_cient background for understanding document clustering.

In this model, each document, dj , is _rst represented as a term-frequency vector in the term-space:

djtf = (tf1j ; tf2j ; :::; tfV j) j = 1; 2; :::;D

where tfij is the frequency of the ith term in document dj , V is the total number of the selected vocabulary, and D is the total number of documents in the collection.

Next, each term is weighted based on its inverse document frequency (IDF). The basic idea is that if a term appears frequently across all documents in a collection, its discriminating power should be discounted.

So, tf -idf vector for each document is obtained

dj = (tf1j _ idf1; tf2j _ idf2; :::; tfV j _ idfV)0 j = 1; 2; :::;D

Usually, in text mining techniques, the term frequency of a term (word or phrase) is computed to explore the importance of the term in the document. However, two terms can have the same frequency in their documents, but one term contributes more to the meaning of its sentences than the other term. It is important to note that extracting the relations between verbs and their arguments in the same sentence has the potential for analyzing terms within a sentence. The information about who is doing what to whom clarifies the contribution of each term in a sentence to the meaning of the main topic of that sentence.

SEMANTIC ROLE LABELING

Semantic role labeling is a task in natural language processing consisting of the detection of the semantic arguments associated with the predicate or verb of a sentence and their classification to their specific roles. Semantic parsing of sentences is an important task toward natural language understanding, and has immediate applications in tasks such information extraction and question answering. In this semantic role labeling (SRL), for each verb in a sentence, the goal is to identify all constituents that fill a semantic role, and to determine their roles, such as Agent, Patient or Instrument, and their adjuncts, such as Locative, Temporal or Manner. For example, given a sentence

"I left my pearls to my daughter-in-law in my will.", the goal is to identify different arguments of the verb left which yields the output:[A0 I] [V left] [A1 my pearls] [A2 to my daughter-in-law] [AM-LOC in my will].Here A0 represents the leaver, A1 represents the thing left, A2 represents the benefactor, AM-LOC is an adjunct indicating the location of the action, and V determines the verb.

A machine learning algorithm for shallow semantic parsing was proposed in[5]. Their algorithm is based on using Support Vector Machines (SVMs) which results in improved performance over that of earlier classifiers by Shallow semantic Shallow semantic parsing is formulated as a multiclass classification problem. SVMs are used to identify the arguments of a given verb in a sentence and classify them by the semantic roles that they play such as AGENT, THEME, and GOAL.

\

CONCEPT-BASED MINING MODEL

In this model instead of clustering the documents based on frequent terms cluster the documents based on the document, and corpus levels.In this model each sentence in the document is labeled automatically based on the PropBank notations. After running the semantic role labeler, each sentence in the document might have one or more labeled verb argument structures. The number model which alyzes the terms on the of generated labeled verb argument structures is entirely dependent on the amount of information in the sentence. The sentence that has many labeled verb argument structures includes many verbs associated with their arguments. The labeled verb argument structures, the output of the role labeling task, are captured and analyzed by the concept-based mining model on sentence, document, and corpus levels.

Sentence-Based Concept Analysis

To analyze each concept at the sentence level, a new concept-based frequency measure, called the conceptual term frequency ctf. The ctf calculations of concept c in sentence s and document d are as follows:

Calculating ctf of Concept c in Sentence s:

The ctf is the number of occurrences of concept c in verb argument structures of sentences. The concept c, which frequently appears in different verb argument structures of the same sentence s, has the principal role of contributing to the meaning of s. In this case, the ctf is a local measure on the sentence level.

Calculating ctf of Concept c in Document d

A concept c can have many ctf values in different sentences in the same document d. Thus, the ctf value of concept c in document d is calculated by:

ctf=ctfn/sn

where sn is the total number of sentences that contain concept c in document d.

Taking the average of the ctf values of concept c in its sentences of document d measures the overall importance of concept c to the meaning of its sentences in document d. A concept, which has ctf values in most of the sentences in a document, has a major contribution to the meaning of its sentences that leads to discover the topic of the document. Thus, calculating the average of the ctf values measures the overall importance of each concept to the semantics of a document through the sentences. To analyze each concept at the document level, the concept based term frequency tf , the number of occurrences of a concept (word or phrase) c in the original document, is calculated. The tf is a local measure on the document level.

CORPUS-BASED CONCEPT ANALYSIS

To extract concepts that can discriminate between documents, the concept-based document frequency df, the number of documents containing concept c, is calculated. The df is a global measure on the corpus level. This measure is used to reward the concepts that only appear in a small number of documents as these concepts can discriminate their documents among others. The process of calculating ctf, tf , and df measures in a corpus is attained by the proposed Algoritm which is called Concept-based Analysis Algorithm.

Concept-Based Analysis Algorithm

```
ddoci is a new Document
L is an empty List (L is a matched concept list)
sdoci is a new sentence in ddoci
Build concepts list Cdoci from sdoci
for each concept ci 2 Ci do
    compute ctfi of ci in ddoci
    compute  tfi of ci in ddoci
    compute dfi of ci in ddoci
    dk is seen document, where k={0; 1; . . . ; doci-1 }
    sk  is a sentence in dk
    Build concepts list Ck from sk
    for each concept cj 2 Ck do
        if (ci == cj) then
            update dfi of ci
            compute ctfweight = avg(ctfi; ctfj)
            add new concept matches to L
        end if
    end for
end for
output the matched concepts list L
```

The concept-based analysis algorithm describes the process of calculating the ctf, tf , and df of the matched concepts in the documents. The concept-based analysis algorithm is capable of matching each concept in a new document d, with all the previously processed documents in o(m) time, where m is the number of concepts in d. After finding the matching concepts among the documents calculate the similarity between the documents based using concept based similarity measure.

Example of Calculating the Proposed

Conceptual Term Frequency (ctf) Measure Consider the following sentence:
Texas and Australia researchers have **created** industry-ready sheets of materials **made** from nanotubes that could **lead** to the development of artificial muscles.In this sentence, the semantic role labeler identifies three target words (verbs), marked by bold, which are the verbs that represent the semantic structure of the meaning of the sentence.

These verbs are created, made, and lead. Each one of these verbs has its own arguments as follows:

- [ARG0 Texas and Australia researchers] have [TARGET created] [ARG1 industry-ready sheets of materials made from nanotubes that could lead to the development of artificial muscles].

- Texas and Australia researchers have created industry-ready sheets of [ARG1 materials] [TARGET made] [ARG2 from nanotubes that could lead to the development of artificial muscles].

- Texas and Australia researchers have created industry-ready sheets of materials made from [ARG1 nanotubes] [R-ARG1 that] [ARGM-MODcould] [TARGET lead] [ARG2 to the development of artificial muscles].

Arguments labels1 are numbered ARG0, ARG1, ARG2, and so on depending on the valency of the verb in sentence. The meaning of each argument label is defined relative to each verb in a lexicon of Frames Files . Despite this generality,ARG0is very consistently assigned an Agent-type meaning, while ARG1 has a Patient or Theme meaning almost as consistently .

Thus, this sentence consists of the following three verb argument structures:

1. First verb argument structure for the verb created:
- [ARG0 Texas and Australia researchers]
- [TARGET created]
- [ARG1 industry-ready sheets of materials made from nanotubes that could lead to the development of artificial muscles].
-

2. Second verb argument structure for the verb made:
- [ARG1 materials]
- [TARGET made]
- [ARG2 from nanotubes that could lead to the development of artificial muscles].
-

3. Third verb argument structure for the verb lead:
- [ARG1 nanotubes]
- [R-ARG1 that]
- [ARGM-MOD could]
- [TARGET lead]
-

A cleaning step is performed to remove stop words that have no significance, and to stem the words using the popular Porter Stemmer algorithm. The terms generated after this step are called concepts. In this example, stop words are removed and concepts are shown without stemming for better readability as follows:

1. Concepts in the first verb argument structure of the verb created:
- Texas Australia researchers
- created

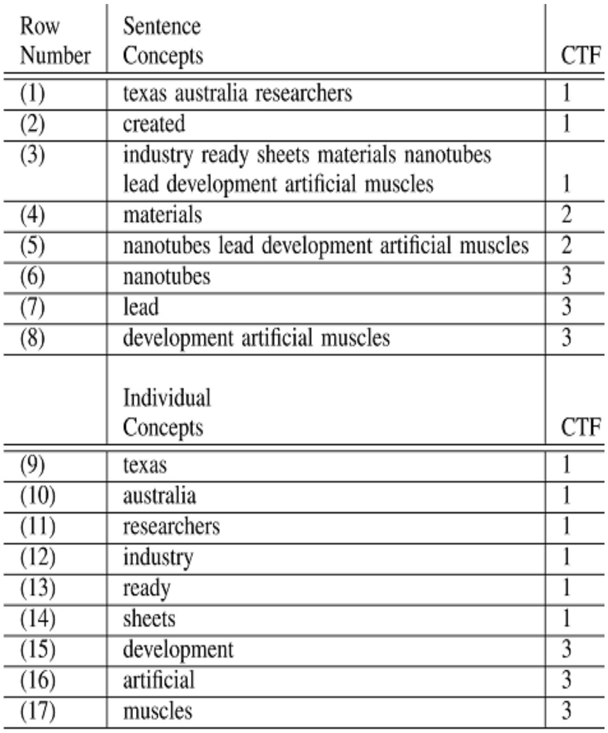
- industry-ready sheets materials nanotubes lead development artificial muscles
-

2. Concepts in the second verb argument structure of the verb made:

- materials
- nanotubes lead development artificial muscles
-

3. Concepts in the third verb argument structure of the verb lead:

- nanotubes
- lead
- development artificial muscles.



Row Number	Sentence Concepts	CTF
(1)	texas australia researchers	1
(2)	created	1
(3)	industry ready sheets materials nanotubes lead development artificial muscles	1
(4)	materials	2
(5)	nanotubes lead development artificial muscles	2
(6)	nanotubes	3
(7)	lead	3
(8)	development artificial muscles	3
	Individual Concepts	CTF
(9)	texas	1
(10)	australia	1
(11)	researchers	1
(12)	industry	1
(13)	ready	1
(14)	sheets	1
(15)	development	3
(16)	artificial	3
(17)	muscles	3

CONCEPT BASED SIMILARITY MEASURE

The concept-based similarity measure relies on three critical aspects.

First, the analyzed labeled terms are the concepts that capture the semantic structure of each sentence. **Second,** the frequency of a concept is used to measure the contribution of the concept to the meaning of the sentence, as well as to the main topics of the document. **Last**, the number of documents that contains the analyzed concepts is used to discriminate among documents in calculating the similarity. The conceptual term frequency (ctf) is an important factor in calculating the concept based similarity measure between documents. The more frequent the concept appears in the verb argument structures of a sentence in a document, the more conceptually similar the documents are.

This similarity measure is a function of the following factors:

1. the number of matching concepts, m, in the verb argument structures in each document d,
2. the total number of sentences, sn, that contain matching concept ci in each document d,
3. the total number of the labeled verb argument structures, v, in each sentence s,
4. the ctfi of each concept ci in s for each document d, where i = 1; 2; . . .;m, as mentioned in
5. the tfi of each concept ci in each document d, where i = 1; 2; . . .;m.
6. the dfi of each concept ci, where i = 1; 2; . . .;m,
7. the length, l, of each concept in the verb argument structure in each document d,
8. the length, Lv, of each verb argument structure which contains a matched concept, and
9. the total number of documents, N, in the corpus.

The concept-based similarity between two documents, d1 and d2 is calculated by:

Simc(d1,d2) = _i=1max(li1/Lvi1 , li2/Lvi2)* weighti1*weighti2

weighti = (tf weighti + ctf weighti)* log(N/dfi)

tf weighti value presents the weight of concept i in document d at the document level. ctf weighti value presents the weight of the concept i in document d at the sentence level based on the contribution of concept i to the semantics of the sentences in d. **log(N/dfi)** value rewards the weight of the concept i on the corpus level, when concept I appears in a small number of documents.

tf weighti = tfij/cn(tfij)2

where cn is the total number of the concepts which has a term frequency value in document d.

ctf weighti = ctfij/cn(ctfij)2

where cn is the total number of concepts which has a conceptual term frequency value in document d.

CLUSTERING ALGORITHM

A *cluster* is a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. Here we use the concept based similarity measure to fine the similarity between the documents. Use any one the existing clustering algorithms to cluster the documents distinguish two types of clustering techniques: *Partitional* and *Hierarchical*

Partitional : Given a database of objects, a partitional clustering algorithm constructs partitions of the data, where each cluster optimizes a clustering criterion, such as the minimization of the *sum of squared distance from the mean* within each cluster.

Different types of partitional clustering algorithms.
1)Single pass clustering algorithm
2)K-means clustering algorithm

Hierarchical : Hierarchical algorithms create a hierarchical decomposition of the objects. They are either *agglomerative* (*bottom-up*) or *divisive* (*top-down*):

Agglomerative **algorithms** start with each object being a separate cluster itself, and successively merge groups according to a similarity measure. The clustering may stop when all objects are in a single group or at any other point the user wants.

Divisive **algorithms** follow the opposite strategy. They start with one group of all objects and successively split groups into smaller ones, until each object falls in one cluster, or as desired. Divisive approaches divide the data objects in disjoint groups at every step, and follow the same pattern until all objects fall into a separate cluster. This is similar to the approach followed by divide-and-conquer algorithms. Most of the times, both approaches suffer from the fact that once a merge or a split is committed, it cannot be undone or refined.

LIMITATIONS

- In concept based mining model outliers may present in the output. due to outliers it is difficult to find subject of specific document.
- It calculates ctf of all concepts present in documents therefore it is time consuming process. To avoid this there is need of selection of top concepts.
-

SOLUTION

The process of selecting the top concepts

- 1. *ddoci is a new Document*
- 2. *L is an empty List (L is a top concept list)*
- 3. for each labeled sentence *s in d do*
- 4. create a *COG for s as in*
- 5. *ci is a new concept in s*
- 6. for each concept *ci in s do*
- 7. compute *tfi of ci in d*
- 8. compute *ctfi of ci in s in d*
- 9. compute *weightstati of concept ci*
- 10. compute *weightCOGi of concept ci based on LCOGi*
- 11. compute *weightcombi = weightstati ¤ weightCOGi*
- 12. add concept *ci to L*
- 13. end for
- 14. end for
- 15. sort *L descendingly based on weightcomb*
- 16. output the *max(weightcomb) from list L*

CONCLUSION

A new concept based mining model composed of four components, is proposed to improve the text clustering quality. By exploiting the semantic structure of the sentences in documents, a better text clustering result is achieved. The first component is the sentence-based concept analysis which analyzes the semantic structure of each sentence to capture the sentence concepts using the proposed conceptual term frequency ctf measure. The second component, document-based concept analysis, analyzes each concept at the document level using the concept-based term frequency tf. The third component analyzes concepts on the corpus level using the document frequency df global measure. The fourth component is the concept-based similarity measure which allows measuring the importance of each concept with respect to the semantics of the sentence, the topic of the document, and the discrimination among documents in a corpus. By combining the factors affecting the weights of concepts on the sentence, document, and corpus levels, a concept-based similarity measure that is capable of the accurate calculation of pair wise documents is devised. This allows performing concept matching and concept-based Similarity calculations among documents in a very robust and accurate way.

REFERENCES

1. An Efficient Concept-Based Mining Model for Enhancing Text Clustering Shady Shehata, Member, IEEE, Fakhri Karray, Senior Member, IEEE, and Mohamed S. Kamel, Fellow, IEEE, OCTOBER 2010.
2. H. Jin, M.-L. Wong, and K.S. Leung, “Scalable Model-Based Clustering for Large Databases Based on Data Summarization,”IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 11,pp. 1710-1719, Nov. 2005.
3. S. Shehata, F. Karray, and M. Kamel, “Enhancing Text ClusteringUsing Concept-Based Mining Model,” Proc. Sixth IEEE Int’l Conf.Data Mining (ICDM), 2006.
4. S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky,“Shallow Semantic Parsing Using Support Vector Machines,”Proc. Human Language Technology/North Am. Assoc. for ComputationalLinguistics (HLT/NAACL), 2004.

5. S. Pradhan, K. Hacioglu, W. Ward, J.H. Martin, and D.Jurafsky, "Semantic Role Parsing: Adding Semantic Structure to Unstructured Text," Proc. Third IEEE Int'l Conf. Data Mining(ICDM), pp. 629-632, 2003.
6. S. Pradhan, K. Hacioglu, V. Krugler, W. Ward, J.H. Martin,and D. Jurafsky, "Support Vector Learning for Semantic Argument Classification," Machine Learning, vol. 60, nos. 1-3,pp. 11-39, 2005