SENTENCE SIMILARITY BASED TEXT SUMMARIZATION USING CLUSTERS

Mohd. Saif Wajid (PG Student)
Department of Computer Science
Babu Banarasi Das University,
Lucknow,India
mohdsaif06@gmail.com

Shivam Maurya(PG Student)
Department of computer science
Babu Banarasi Das University
Lucknow , India
reverentshivam@gmail.com

Mr. Ramesh Vaishya
Sr. Lecturer ,Department of
Computer Science
Babu Banarasi Das National Institute
of Technology and Management
Lucknow.India
bbnitm.rv@gmail.com

ABSTRACT-

The computer is based on natural language on summarization and machine system. It is very difficult for human being manually summarize large amount of text. The greatest challenge for text summarization to summarize convent from number of textual and semi structured sources, including text , HTML page, portable document file . This summarization can be determine from internal and external measure. Our proposed work sentence similarity based text summarization using clusters help in finding subjective question and answer on internet .This work provide short units of text that belongs to similar information. I proposed my work on sentence similarity based computation that help to experiment for similar text computation. Extractive summarization text system choosing a subset of similar group from the text . proposal work i used the part of speech , proper noun, verb, pronouns such as he, she, and they etc. With the help of part of speech we find important sentence using statistical method like proper noun and sentence similarity system .It based on internet information that that contain picky sentence.

Index Terms- Similarity Computation ,Primitive Extraction ,Merging similarity, Clustering Techniques, Compute text similarity.

Introduction-

one approach to sentence similarity based text summarization using clusters for summarizing has proved efficiency and gained popularity is similarity based summarization .The principle behind similarity based summarization is that important in information is repeated in different sentence on the same event .Identifying this repeated important text Summarization is one important approach to managing the large amount of text . Summarization can diminish the amount of text document is relevant to their information need. Summarizing text by shortening a long document to present the document's content in same event. As progress was made in single document summarization, researchers began to study sentence similarity based summarization. Numbers of documents on the same event reporting on developments in the same court case. The goal is to produce a short summary that gives an overview of all the documents. One approach to similar summarization that has proven gained popularity is similarity-based summarization. We ranked each sentences based on their feature and use manually summarized data for calculation of weight of each feature. We also use chart theoretic link diminution technique called threshold scale techniques. The text is represent as a chart with individual sentences as the nodes and similarity between the sentences as the weights on the links. To calculate similarity between the sentences it is necessary to represent the sentences as vectors of terms. The sentences are selected for inclusion in the final summary on the basis of their relative importance in the graph and feature score in the text. What is important can depend upon the user needs or the purpose of the summary.

Similar sentences are based on each feature, and it combines all the similarities into a single similarity value representing the overall similarity of the two sentences taking example -in the following two sample sentences primitives. This construction has allowed me to experiment with similar combinations of primitives and translation method-

Sentence 1- The teacher ran the program.

Sentence 2-The runner ran the race very shortly.

The noun primitives -
Sentence 1- are (teacher, program) .
Sentence 2- are (runner, race).
The verb primitive in both sentences is (ran).

Two features, verb similarity and noun similarity, are computed over the two primitive types, and while similarity is high over the verb feature they both share the same and only verb it is low over the noun feature.

Similarity text summarization-

Similar text summarization corresponds to the process in which a computer creates a zipped version of the novel text (a collection of texts) still preserving most of the information present in the original text. This process can be seen as density and it inevitably suffers from information loss. Simpler approaches were then explored that consist of extracting representative text-span, using arithmetic techniques or the techniques based on surface realm sovereign analyses. This is typically done by position document sentences and selecting those with higher score and minimum overlie. Thus a similar text summarization system must identify important parts and protect them. Similarity based summarization approaches are not new in the area of summarization, similarity based summarization is an accepted, similarity-based approaches, they are usually applied to similar text summarization system.

CONTRIBUTIONS-

Flexible framework for sentence similar text trialling-

Sentence similarity text summarization supports rapid development of features for similarity computation for same event, and support for similar translation mechanisms over those.

Trialling with and evaluation of event levels of translation for similar text similarity recognition-

Similar text summarization can be used at same levels for similar text similarity recognition. I have compared full document translation using machine translation systems to primitives level translation that translates at the word level and translation of phrases extracted from the papers.

Methods that are easily portable to new events-

Using phrases and primitives for translation from large collections of text and their translations allows one to quickly add support for similar sentences.

Investigating primitives for similarity and translating primitives-

An original contribution of this work is the study of primitives that are compatible across sentence for the similarity computation process and methods of translating those primitives. Similarity totalling performed over primitives and their translations extracted from the native sentence is more easily extensible to sentence for which we do not already have a full machine translation system. For high precision responsibilities requiring identification of sentences, translation at the primitive level performs better than similarity computation using machine translated input documents. In this work, I investigate word-level primitives, and named entity based noun phrase primitives for similarity computation between similar events.

Extractive Similar Summarization-

Similar Sentence based sentence summarization techniques are commonly used in sentence similarity summarization. The summary bent by the summarizer is a subset of the original text Extractive similar summarizer chosen out the most germane sentences in the document with maintaining the low severance in the summary . In this work the extraction unit is defined as a sentence. Sentences are well clear linguistic entities and have self enclosed meaning. So the aim of an extractive summarization system becomes, to identify the most imperative sentences in a text. The theory behind such a system is that there exists a rift of sentences that present all the category point of the text. In this case the general framework of an summarize extractive summarization works by ranking creature sentences. Most of the extractive summarization systems differ in this juncture. A sentence can be ranked using a clue indicating its significance in the text. There are different matrix for sentence choice from the text to produce summary. It is a task of classification of sentence which are defined in figure no. 1 -

1-Sentence boundary inequity
2-structure words of the contents
3-Calculation of sentence importance (ranking)
4-Selection of ranked sentences

Figure no.1-Framework of sentence similar text summarization system.

LITERATURE REVIEW-

In the previous work of English documents indicating similarities and differences between. The ability to indicate differences between the document sources is a novel contribution, as previous work focused on identifying similarities between documents. This work leads the way for further research in active analysis of difference in perspectives between documents sets and languages, a boon for information analysts.

Applying sentence text similarity computation-

This work presents two summarization systems that use text similarity. The first uses similarity to replace machine translated sentences from text documents with similar sentences to recover readability of summaries. The steps in the pre-processing stage are to segment the text of the papers into units to compare for similarity, and to create unusual representations of the text, such as part of words tagged versions, that will be used in stages to extract primitives

Primitive Extraction-

This research paper define similarity between two units, we need to identify the tiny elements used to compute similarity. These are called primitives. Primitives are common module (for example, all stem words, all nouns, all noun phrases), while a particular instance of a primitive would be a A specific word, or a specific noun phrase. Similarity between two units is computed using quality over these primitives, which will be discussed shortly. The second stage identities and extracts primitives for each unit. Primitive extractors are defined on a per-language basis using a plug-in design making it easy to add support for different languages by simply creating primitive extractors for that language.

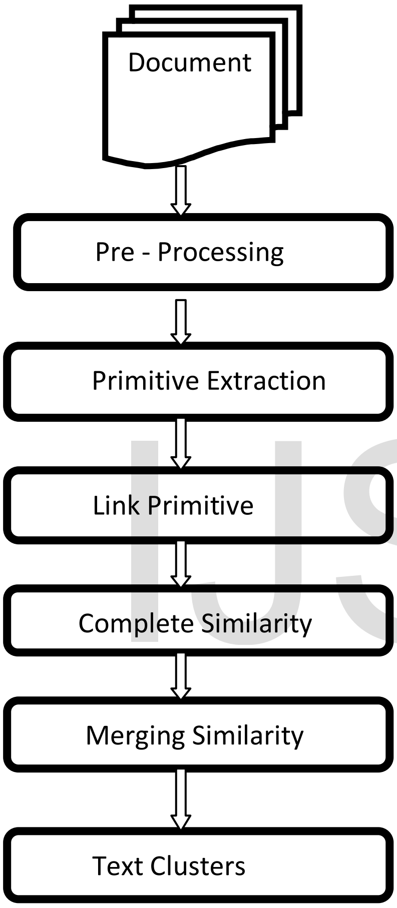

Figure no.2-Similarity finder architecture.

Primitive Linking-

The primitive linking phase is not a full translation phase. Since the goal is to use the translations to tie other potentially related primitives, I prefer to blunder on the side of Opportunistically linking two primitives even if there might only be a feeble relationship between them. Since there is at least one primitive for each token in a sentence, there are often a large number of primitives to compare between two sentences. Sentences that are similar generally have more than solitary link between translated primitives suitable to additional links from other related words in the sentence.

Similarity Computation-

Similarity between two units is computed on many sort defined over the primitives Identified for each unit. Before play the genuine comparison between the units, the units which should be compared are identified. Similarity finder uses an approach that avoids comparing units that will not be found to be similar. To collect units to compare, a primitive is elected from the primitive tracking data structure and all units containing the primitive or a linked primitive are compared against each other. An N × N array, where N is the number of text units, tracks which units have been compared, ensuring that similarity is computed only once for each pair of units.

The similarity of two units, U1 and U2 with primitives P1 and P2, with the strength of a link between primitive P1a and P2b given as WP1a;P2b is determined as-

$$S_{U_1,U_2} = \frac{\sum_{a=1}^{|P_1|}\sum_{b=1}^{|P_2|}(W_{P_{1a},P_{2b}})}{|P_1 \cup P_2|}$$

Merging similarity-

The similarity computation process used in Similarity creates a similarity matrix between the units on several dimensions. For each of the primitives extracted from the units, a feature comparator is used to compare the similarity of the two units over that primitive. The similarity computation point results in a N× *N* ×F similarity matrix, where N is the number of textual units, and F is the number of features that were used during the run Before clustering the units, the N ×N × F feature similarity medium is converted into a N ×N matrix such that each element contains a single charge expressing the total similarity between the two units.

Clustering Techniques

Sentence similarity uses clustering in two ways-

- Document clustering
- clustering text units

Cluster analysis is a general technique for multivariate analysis that assigns items to groups automatically based on a similarity computation. Cluster analysis has been applied to Information Retrieval to provide more efficient or more effective retrieval, and to structure large sets of retrieved documents. When applying clustering to text documents, the attributes over which the clustering is performed and their representation must be selected, and a clustering method and similarity measure must be chosen. The estimation is based on the similarity values between the written units. The number of clusters c for a set of n textual units in m connected components is determined by-

$$c = m + \left(\frac{n}{2} - m\right)\left(1 - \frac{\log(L)}{\log(P)}\right)$$

where L is the observed number of links between units based on their similarity, and

$$P\ (= n(n-1)/2)$$

is the maximum possible number of links. A non-linear interpolating function is used to account for the fact that usually L ≤ P.

Preprocessing-

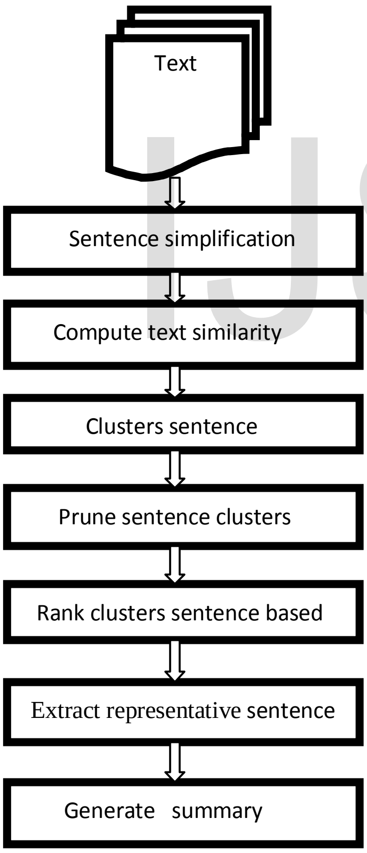

Figure no.3- CAPS System Architecture of Preprocessing.

The first module is a pre-processing module, which prepares the input articles for processing I have designed a sentence independent that abstracts the universal pre-processing steps for-

Next, sentence clusters are partitioned by source, resulting in multiple clusters of similar sentences from English sources, multiple clusters of sentences from multiple clusters of sentences from both English. Finally, I rank the sentences in each source partition using a TF*IDF the ranking determines which clusters contribute to the summary (clusters below a threshold are not included) as well as the ordering of sentences. For each cluster, we extract a representative sentence (note that this may be only a portion of an input sentence) to form the summary. In this section, I describe each of these stages in more detail.

Sentence simplification-

As with the summarization system presented it is possible to performing syntactic sentence simplification on the input English text. I have previously performed experiments using both perform syntactic simplification and not using simplification on the input English text, and show the results. syntactic sentence simplification with this system as well because it allows one to measure similarity otherwise be possible. I use a sentence simplification system developed at Cambridge University for the task. The generated summary often includes only a portion of the un simplified sentence, thus saving space and improving accuracy. I opt to use syntactic sentence simplification only instead of using syntactic simplification with pronoun resolution. The pronoun resolution phase included in the software sometimes makes anaphoric reference resolution errors, resulting in incorrect re-wordings of the text.

Compute text similarity-

Text similarity sentences is computed using similarity a program I developed which uses simple feature identification and translation at word and phrase levels to generate similarity scores between sentences across. Text similarity between manual or machine translated English documents and English is computed with Similarity an English-specific program for text similarity computation that similarity was model after. similarity for English is presented in I present a third baseline approach using the cosine distance for text similarity computation.

Sentence clustering and pruning-

Sentence clustering uses the same clustering component. Each cluster represents a fact which can be added to the summary each sentence in the generated summary corresponds to a single cluster. Since every sentence must be included in some cluster, individual clusters often contain some sentences that are not highly similar to others in the cluster. To ensure that our clusters contain sentences that are truly similar, I implemented a cluster pruning stage that removes sentences that are not very similar to other sentences in the cluster This pruning step ensures that all sentences in a sentence cluster are similar to every other sentence in the cluster with a similarity above a given similarity threshold. I illustrate the procedure with the following example. For the cluster with these initial sentences Based on the similarity values between the sentences in the cluster, those sentences that have values lower than the threshold are removed The cluster is then The resulting cluster contains sentences that are much more similar to each other, which is important for my summarization strategy since I select a representative sentence from each cluster that is included in the summary. I do not want to make sentences that are not representative of the cluster available for inclusion in the summary.

Ranking clusters-

Once the clusters are partitioned by language, CAPS must determine which clusters are most important and should be included in the summary. Typically, there will be many more clusters than cannot in a single summary average input data set size is 7263 words, with an average of 4050 words in clusters, and I am testing with 800 word summaries, 10% of the original text. In the default arrangement, CAPS uses TF*IDF to rank the clusters; those clusters that contain words that are most unique to the current set of input documents are likely to present new, important information. For each of the three types of sentence clusters. The TF*IDF score for a cluster is the sum of all the term frequencies in the sentences in the cluster multiplied by the inverse document frequency of the terms to discount frequently occurring terms, normalized by the number of terms in the cluster. The inverse document frequencies are computed from a large corpus of AP and Reuters news. CAPS has two other measures for ranking clusters: the number of unique sentences in each cluster, and the number of unique sentences in a cluster weighted by the TF*IDF score of the cluster. Experimentation over a single test document set showed that the TF*IDF score performed best of the three, and results from this thesis use that cluster ranking method When using text in the input and text similarity computation phases, the text is translated into similar after the clustering phase. TF*IDF counts are computed over the machine translated text. This is done because the ranking of clusters has to be done over English, and mixed clusters, which presents a problem: how to rank the Arabic and mixed clusters? For Arabic-only clusters, a TF*IDF move towards using IDF values from a large Arabic corpus could be used, but it is unclear if direct application of TF*IDF to clusters with both languages and diverse IDF values for each languages would be applicable. As the Arabic sentences need to be translated for presentation in an English summary anyway, and many of the sentences have been dropped through the clustering and pruning process, machine translation is performed at this step, and clusters are ranked with the machine translated versions of the sentences.

Sentence selection-

The cluster ranking phase determines the order in which clusters should be included in the summary. Each cluster contains several sentences, but only one of these is selected to represent the cluster in the summary.

There are three methods implemented to select a exact sentence to represent the cluster-

1. The sentence most similar to all other sentences based on the computed similarity values.

2. A TF*IDF based ranking method that selects a sentence with the highest TF*IDF score.

3. A method that constructs a centroid sentence in a vector space model, and selects the most similar sentence to the centroid To compute a TF*IDF score for clusters with text in multiple languages, one must have a (preferably large) corpus to derive IDF values for terms Experimentation over a test set showed that the first method performed best, so that is the method used in these experiments.

Only the set of unique sentences are evaluated for each cluster. In this sort of task, many of the input documents repeat text verbatim, as the documents are based on the same newswire (Associated Press, Reuters, etc.) report, or are updated versions of an earlier report. In order to avoid giving undue weight to a sentence that is repeated multiple times in a cluster, the unique sentences in each cluster are first recognized Unique sentences are recognized using a simple hash function, removing leading and trailing white space.
Similarity based selection: To select a sentence based on the text similarity values first the set of unique sentences is determined as described above. For each unique sentence in the cluster, its average similarity to every other unique sentence in the cluster is computed.

The unique sentence with the highest average similarity is then chosen to represent the cluster centroid sentence is computed, and the closest single sentence is chosen to represent the cluster. In order to generate a digest, CAPS draws from the English sources as mochas possible. For summary sentences from clusters with only Arabic sentences, clearly nothing can be done to improve upon the machine translated Arabic. But when generating the summary from mixed English clusters, CAPS uses English phrases in place when the similarity value is above a learned threshold, a is the case for the pruned clusters. this method improved summary quality in68% of the cases in a human study.

Summary generation-

Once the clusters are ranked and a sentence has been selected to represent each cluster, the main remaining issue is how many sentences to select for each partition There are two parameters that control summary generation total summary
Word limit, and the number of sentences for each of the three partitions. The system takes sentences in proportions equal to the relative partition sizes. For example, if CAPS generates clusters, English clusters, clusters, then the ratio of sentences from each partition is English. The smallest partition size is divided through the to determine the ratio. The total word count is divided among partitions using this ratio.

Acknowledgement-

We sincerely thanks to every referees and scientific experts for their valuable observations and suggestions in the preparation of the present piece of writing. Our team members and teachers for their sustain from the beginning of the concept till the achievement of this research paper.

Conclusions-

I have presented a system for generating English summaries of a set of text documents on the same event, where the text documents are drawn from English Unlike previous summarization systems, CAPS explicitly identifies agreements and similarity between English. It uses sentence overview and similarity scores to identify when the same facts are presented in two similar sentences, and clustering to group together all sentences . I presented an evaluation methodology to measure accuracy of CAPS partition of similar facts. The evaluation shows that our similarity metric outperforms a baseline metric for identifying clusters based on English sentence and performs almost as well using machine translated text as manual translations for identifying important content exclusive similar clusters.

Future work-

Further integration of statistical machine translation methods-

A distortion model might help improve Similarity results at discover sentences that are translations of each other. similar sentences that might not be translations of each other conveying the exact same information, a distortion model might impose too many restrictions, giving similar, but structurally similar sentences, low probabilities.

Noun Phrase Variant recognition-

Noun phrase variant recognition is an area where better translation methods would help. Given a feature that extracts noun phrases in similar to properly match to a noun phrase in another sentence would require either a translation mechanism that produces an N-best list with all likely variants of a noun phrase, or a noun phrase variation system. This section describes some related work in noun phrase variant recognition, and early experiments I performed with Sentence similarity and noun phrase variation in English. Initial results were not encouraging, and I consider a more in-depth search is required to see improvement based on these techniques.

Noun Phrase Variation-

One of the early areas of this thesis work was the investigation of using noun phrase variation to recognize different forms of noun phrases across documents and across languages. Noun phrase variation was used by Bourigault 1992 for the identification of terminological units. Maximal length noun phrases were known and parsed to identify likely terminological units due to the grammatical structure of the noun phrases. The resulting terminological units were then passed to a human expert for support.

References-

Masayuki Asahara and Yuji Matsumoto. Extended models and tools for high performance part-of-speech tagger. In Proceedings of COLING 2000, July 2000.

Regina Barzilay. Information Fusion for Multidocument Summarization: Para- phrasing and Generation. PhD thesis, Columbia University, 2003.

BBN. Bbn identifinder http://www.bbn.com/, 2004.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin.
A statistical approach to machine translation. Computational Linguistics, 16(2):79{85, 1990.

Sasha Blair-Goldensohn, Kathleen McKeown, and Andrew Hazen Schlaikjer. A hybrid approach for qa track definitional questions. In 12th Text Retrieval Conference (TREC 2003), Gaithersburg, MD, November 2003.

Sasha Blair-Goldensohn, Kathleen R. McKeown, and Andrew Hazen Schlaikjer. Answering Definitional Questions: A Hybrid Approach, chapter 4. AAAI Press, 2004.

Regina Barzilay, Kathy McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In Proceedings of the 37th Association of Computational Linguistics, Maryland, June 1999.

Didier Bourifault. Surface grammatical analysis for the extraction of terminological noun phrases. In Proceedings of the 14th International Conference on Computational Linguistics, pages 977{981, 1992.

Christopher Buckley. Implementation of the smart information retreival system. Technical Report Technical Report 85-686, Cornell University, Ithaca, New York, 1985.

T. Buckwalter. Buckwalter arabic morphological analyzer version 1.0, linguistic data consortium (ldc) catalog number ldc2002l49 and isbn 1-58563-257-0., 2002.

Aitao Chen, Fred Gey, and Hailing Jiang. Alignment of English parallel corpora and its use in cross-language information retrieval. In Proceedings of the 19th International Conference on Computer Processing of Oriental Languages, Seoul, Korea, May 2001.

N. Collier and H. Hirakawa. Acquisition of english proper nouns from noisy-parallel newswire articles using katakana matching. In Proceedings of the Natural Language Pacific Rim Symposium (NLPRS-97), Phuket, Thailand, December 1997.

Aitao Chen. Cross-language retrieval experiments at clef 2002.

In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, CLEF, volume 2785 of Lecture Notes in Computer Science, pages 28{48. Springer, 2002.

Hsin-Hsi Chen and Chuan-Jie Lin. A multilingual news summarizer. In Pro-ceedings of the 18th International Conference on Computational Linguistics, pages 159 {165, 2000.

J. Cohen. A coe_cient of agreement for nominal scales. Educational and Psy- chological Measurement, 20:37{46, 1960.

William W. Cohen. Learning trees and rules with set-valued features. In AAAI/IAAI, Vol. 1, pages 709{716, 1996.

Ido Dagan, Alon Itai, and Ulrike Schwall. Two languages are more informative than one. In Proceedings of the 29th conference on Association for Computational Linguistics, pages 130{137. Association for Computational Linguistics, 1991.

Nina Wacholder David Kirk Evans, Judith L. Klavans. Document processing with linkit, April 2000.