International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 12, Number 23 (2017) pp. 13040–13048
© Research India Publications, http://www.ripublication.com

Automatic Summarization for Agriculture Article

Laxmi B. Rananavare
Research Scholar, Dept of Computer Science and Engg, Sri Venkateswara University College of Engg,
Tirupathi, Andhra Pradesh, India-517502
Faculty Member in REVA University, Bangalore-560064
ORCID ID: 0000-0001-5517-5084

P. Venkata Subba Reddy
Professor, Dept of Computer Science and Engg,
Sri Venkateswara University College of Engg, Tirupathi, Andhra Pradesh, India-517502.

Abstract: In India, agriculture is the back-bone of economy. There are various means to harvest resources for food, clothing and dwelling. Knowing only the basics of agriculture is not enough, one should be aware of agriculture practices throughout the world. New technologies and researches in agriculture field would benefit development. Lot of e-information about agriculture field is available. When reader or user browses agriculture information it is quite difficult for the reader or farmer to go through each every information. Some of the information is redundant on different websites. The basic issue to be addressed here is to ease and automate the process of collection and processing the vast data available through trustworthy news sources. This work proposes a agriculture article summarization system which access information from various on-line articles automatically and summarizes information. Devising an application which automatically collects digital on-line news articles based on a key-word from local newspaper articles and then summarizes them using Natural Language Processing techniques to semantically cluster sentences and the extract sentences from said clusters using Centroid-Based sentence summarization techniques.

Keywords:

INTRODUCTION

Nowadays the volume of data in electronic form is increasing rapidly. It can be structured data like databases, company legacy data; or unstructured data like text, images etc. According to [McKnight, 2005] between 85% and 90% of data is held in unstructured form. Therefore, text mining is necessary for managing and extracting useful information from unstructured sets of data, such as web-pages, news reports and emails, using a variety of text mining techniques. Hence, text mining has become an important and active research eld. It is well known that text mining techniques have mostly been developed for the English language because most electronic data is in English. Using this to our advantage, it is an obvious next step to employ these techniques for sifting through the multitude of available online data to mine facts and figures from various sources and then summarize them efficiently to use in tracking various events in and around an area under Indian farming.

In this paper information extraction based on computational linguistic techniques are used to summarize the text where the summarization is considered as three prominent processes 1) filtering (by removing stop words), 2) highlighting 3) organizing information from different logically related text sources thereby producing a short meaningful summary of the text information spread across pages of texts.

Key tasks in Summarization

1. Automatically extract on-line articles from agriculture websites based on a key-word.
2. Divide entire articles as a group of sentences, which acts as the data-set for further processing.
3. Representing sentences in a machine readable and understandable format.
4. Detecting semantic similarity between sentences so as to eliminate factual redundancy in summary.
5. Clustering similar sentences to distinguish between semantically different sentences.
6. Picking sentences among-st clusters which represent the information presented by the corresponding cluster.
7. Arranging the sentences chronologically to display the developments as they happened.

LITERATURE SURVEY

Summarization of documents has garnered a lot of traction and interest in the recent years. The main reason there is a necessity for summarization is to scale through the multitude of information present on the internet and then present it in a concise format which stays true to the subject at hand. There are mainly two different categories to summarization.

In it, there are two ways to extract summaries viz.

1. Abstractive summaries produce generated text from the important parts of the documents, usually by giving a certain amount of hard-coded sentence structures.
2. Extractive summaries identify important sections of the text and use them in the summary as they are.

Summarization of Single Document

Summarization of single news documents consists of shortening of facts presented by single page news articles. Usually, the ow of information in a given document. is not uniform, which means that some parts are more important than others. The major challenge in summarization lies in distinguishing the more informative parts of a document from the less ones. Though there have been instances of research describing the automatic creation of abstracts, most work presented in the literature relies on verbatim extraction of sentences to address the problem of single-document summarization. Most Single Document summaries are extractive in nature [1]. There are mainly two methods for Single Document Summarization.

1. Machine Learning Statistical Models
2. Deep Language Analysis Models.

Machine Learning Models

In the late 90s, with the advent of machine learning techniques in NLP, there was a series of publications which involved usage of various statistical models which enabled the deduce whether to extract a given sentence or not. Some techniques are described as follows.

Bayesian Networks were used to classify whether a sentence is worthy of being extracted into the summary, by using human-generated summaries as ground truth [2].

Decision Trees were also used for the same purpose instead of using Naive-Bayes classifiers with the addition of extra features like query signature which scored the sentences based on the number of query words they contained and signature words which were most prevalent in the document etc. [3].

Hidden Markov Models: The basic motivation for using a sequential model is to account for local dependencies between sentences. features like position of the sentence in the documents, number of terms in the sentence, and like-liness of the sentence terms given the document terms [4].

Log-Liner models were improved versions of the Bayesian Networks models [5].

Neural Networks like RankNet we used to rank input sentences of the articles by using the gradient descent method for training [6], [7].

Deep Language Analysis Models

This technique mainly involves

1. Selecting a set of candidate words
2. Finding the chain of relatedness by using various features like WordNet Distance

Inserting relevant missing words using said chains.

Many Works were published with this base technique with varying degrees of success [8] [9], [10].

Advantages and Disadvantages

ADVANTAGES

- Usually keep the original sentence order of the document intact.
- Easier to make than Multi-Document Summaries
- Require less data for accurate results.

DISADVANTAGES

- Most News Articles have Multi-document characteristics.
- Does not allow for a multitude of facts being considered.
- News Articles are already concise and hence, summarization of single documents doesn't necessarily give a good coverage outlook of the query in question.
- Usually not suitable for query based summarization.

Multi-Document Summarization

Extraction of a single summary from multiple documents has gained interest since mid 1990s, most applications being in the domain of news articles. Several Web-based news clustering systems were inspired by research on multi-document summarization, for example Columbia News Blaster, or News In Essence.

This is different from single-document summarization since the problem involves multiple sources of information that overlap and supplement each other, and removal of redundant facts which are presented in a semantically similar but grammatically different structure. The key is not only identifying and coping with redundancy across documents, but also recognizing novelty and ensuring that the nal summary is both coherent and complete.

Various approaches in Multi-Document Summarization are as follows:

- Abstraction and Information Fusion: The summaries are created by merging facts from various document sources to generate an informational summary of the same. These techniques also employed the use of a linguistic generator to create sentences out of words selected based on statistical analysis techniques like TF-IDF scores, noun pronoun and verb weights etc.[11].
- Topic Driven Summarization: Here, a summary containing the information most relevant to a user's information need is produced from a set of topic-related documents. This can be done by employing weighted keyword analysis [12], topic signatures[3] and Statistical methods like Latent Dirichlet Allocation [13] Latent Semantic Indexing and Probabilistic Latent Semantic Analysis [14].
- Graph Based Summarization:Graph and ontology based methods usu-ally use fuzzy logic to determine which of the data is relevant to each other to avoid redundancy in summarization or by supervised learning approaches by guiding the system to learn how to select the correct sentences for summarization [Massandy and Khodra, 2014]. Using classifiers, sentences are also picked from the document Semantic Graph. A document is represented as a graph and each node represents the occurrence of a single word (i.e., one word together with its position in the text) [15].
- Centroid Based Summarization: These techniques use clustering of sentences and then using centroids of said clusters to generate informative summaries [16]. These techniques do not employ a language generation module, thus making it easy to scale and remain domain independent.

ADVANTAGES:

- Multi-document summarization creates information reports that are both concise and comprehensive. With different opinions being put together and outlined, every topic is described from multiple perspectives within a single document.
- Automatic summaries present information extracted from multiple sources algorithmic ally, without any editorial touch or subjective human intervention, thus making it completely unbiased.
- The large amount of data available about a topic is concisely presented and thus makes it easier to study and remain informed.

DISADVANTAGES:

- The need to eliminate redundancy of data.
- Sometimes multiple facts are contradictory(death toll, time and date etc.)

MOTIVATION

A major challenge facing all farming is accurately and efficiently analyzing the growing volumes of data. Data mining is a powerful tool that enables criminal investigators who may lack extensive training as data analysts to explore large databases quickly and efficiently. Computers can process thousands of instructions in seconds, saving precious time. In addition, installing and running software often costs less than hiring and training personnel. Computers are also less prone to errors than human investigators, especially those who work long hours. Newspaper articles offer a broad insight into the daily lives and activities of various crops, weather forecasting or marketing which the farmer usually follow. To track these events, they manually have to follow newspaper cut outs and then make notes and gather intelligence in that fashion. A lot of information can also be lost in the manual collection of such information. Hence,there is a need to develop a system that automatically summarizes the activities of such elements and then gives a visual dashboard to across the information gleaned from processing the articles.The basic issue to be addressed here is to ease and automate the process of collection and processing the vast data available through reputable news sources which particularly target Indian farming. The reason such an application is needed is to collate and collect in-formation from local agricultural news articles and then summarize the same by using unsupervised clustering techniques.

Problem Statement

Devising an application which automatically collects digital on-line news articles based on a key-word from local newspaper articles and then summarizes them using Natural Language Processing techniques to semantically cluster sentences and the extract sentences from said clusters using Centroid-Based sentence summarization techniques.

Work Objectives

1. Collecting data for building a corpora of newspaper articles.
2. Develop a mechanism for breaking down actual sentences into a representative format for semantic analysis.
3. Clustering semantically similar sentences to derive non-redundant summaries of the same.
4. Evaluating the summaries generated by the system.

METHODOLOGY

The user will be first asked to enter a query regarding which the summary has to be generated. The user the selects the newspapers he/she wants to collect the articles from. The articles are the extracted by web crawling and scraping and then saved in the system as archives and as the data-set for summarization.

This Dataset is the further divided into individual sentences and the triplet ex-traction algorithm is run on each sentence and the result is saved for creating of the similarity matrix. Once the similarity matrix is created, a clustering algorithm is applied on it and we get a cluster of all similar sentences. A sentence is selected from each cluster and then put in the summary based on heuristics. Once the summary is created, it is displayed and the user also has an option of picking the sentences which are more well suited for their needs as this information can then be used to tailor a summary engine which uses supervised machine learning techniques.

SUMMARY EXTRACTION

The steps involved in actually deriving the summary from the articles collected from the actual query submitted by the user. There are various steps involved in the process of summarization which can be outlined as follows:

- Breaking down all the articles into individual sentences.
- Breaking down individual sentences into the RDF Triplet format or the Subject Verb Object triplet format for semantic analysis.
- Named Entity Recognition and Stemming.
- Calculating semantic similarities between triplets. Clustering Triplets from Similarity Matrix
- Sentence Selection.

Each of these steps will be explained in detail in the sections to follow.

Sentence Tokenization

Sentence tokenization refers to the practice of dividing a text into a group of sentences. A news article as a whole is basically a collection of interrelated sentences. Since the structure of a news article is usually rigid and uniform [17], it is easier and computationally efficient to parse through the article sentence-wise instead of treating it as one single entity. It is both memory as well as time consuming to semantically analyze the entire article and hence it was decided to treat the entire event which is queried by the user to be treated to have a group of sentences comprising all news articles as the base data-set. The task then became to summarize the article information from the group of sentences rather than per article basis.

It is difficult to understand the semantics automatically from an entire article and hence it is necessary to break down the article into a set of sentences. This is done by the using the Punkt sentences tokenizer in NLTK tool-kit. It is an implementation of the Unsupervised Multilingual Sentence Boundary Detection Algorithm designed in [18]. They proposed to approach sentence boundary detection by first determining possible abbreviations in the text. They do so by identifying three major characteristics of abbreviations.

- An abbreviation is rather compact i.e. there is a close bond between a period and the letter preceding it.
- Abbreviations tend to be short.
- Experimental characterization of internal periods in abbreviations.

Using such heuristics, they built a classifier which determined whether a period was after the end of a sentence or followed preceded by and abbreviation, initial or an ordinal with 99.2% accuracy. Using this model, we can divide the article into a group of sentences which then acts as the base data-set to glean information about the article.

Triplet Extraction

A Triplet is defined in a sentence as a relation between subject and object, the relation being the predicate. The aim here is to extract sets of the form subject, predicate, object out of syntactically parsed sentences. Basically a triplet is used to give an exact semantic sense of what a sentence is talking about. Instead of using the whole sentence to derive meaning; a triplet just uses three words to determine what the sentence is talking about.

Automated summarization is often approached in two phases.
First, key textual elements, e.g., keywords, concepts, and concept relations, are extracted from the text using linguistic and statistical analysis [19]. These are then used to select sentences from the text, enforcing various requirements on coverage and coherence of extracts [20].

The idea of using triples as semantic units for representing content of web documents is well studied in RDF) [21] in the Semantic Web Community.

To begin with; the sentence is first parsed to understand it's grammar by using the Stanford Treebank Parser. Stanford Parser is a natural language parser developed by Dan Klein and Christopher Manning from The Stanford NLP Group [22]. The package contains a Java implementation of the Treebank parser; a graphical user interface is also available, for parse tree visualization called Stanford Tregex. A treebank is a text corpus where each sentence belonging to the corpus has a syntactic structure added to it. In a treebank parser, A sentence (S) is represented by the parser as a tree having three children: a noun phrase (NP), a verbal phrase (VP) and the period (.). The root of the tree will be S. Triplet Extraction is done as follows:

- To find the subject of the sentence, we apply a Breadth First Search in the NP sub-tree and select the rst descendant of NP that is a noun.
- To find the predicate of the sentence, we search for the deepest verb descendant in VP and assign that as the predicate.
- To find objects we search in three different sub-trees. The sub-trees are: PP (prepositional phrase), NP and ADJP (adjective phrase). In NP and PP we search for the first noun, while in ADJP we find the first adjective.

```
Algorithm 1: Triplet Extraction
Data: sentence
Result: A solution or a failure
result ← EXTRACT SUBJECT(NP subtree) ∪
    EXTRACT PREDICATE(VP subtree) ∪
    EXTRACT OBJECT(VP subtree)
if result ≠ failure then
    return result
else
    return failure
end
```

```
Algorithm 2: EXTRACT ATTRIBUTES
Data: word
Result: A solution or a failure
/* search among the word's siblings */
    if adjective(word) then
    result $\leftarrow$ all RB siblings
    else
    if noun(word) then
    result $\leftarrow$ all JJ, ADJP, NP siblings
    else
    if verb(word) then
    result $\leftarrow$ all ADVP siblings
    end
end
end
/* search among word's immediate ancestor siblings */
if noun(word) OR adjective(word) then
    if uncle = PP then
    result $\leftarrow$ uncle subtree

    end
else
    if verb(word) AND (uncle = verb) then
    result $\leftarrow$ uncle subtree
    end
    end
    if result $\neq$ failure then
    return result
    else
    return failure
    end
```

Algorithm 3: EXTRACT SUBJECT

```
Data: NP subtree
Result: A solution or a failure
subject first noun found in NP subtree;
subjectAttributes
EXTRACT ATTRIBUTES(subject);
result subject $\cup$ subjectAttributes;
    if result $\neq$ failure then
    return result
else
return failure
end
```

Algorithm 4: EXTRACT PREDICATE

```
Data: VP subtree
Result: A solution or a failure
predicate deepest verb found in VP subtree;
predicateAttributes
EXTRACT ATTRIBUTES(predicate);
result$\leftarrow$ predicate $\cup$ predicateAttributes;
    if result $\neq$ failure then
    return result
else
    return failure
end
```

Algorithm 5: EXTRACT OBJECT

```
Data: VP subtree
Result: A solution or a failure
for each value in siblings do
if value = NP or PP then
    object $\leftarrow$ first noun in value
else
    object first adjective in value;
objectAttributes $\leftarrow$ EXTRACT ATTRIBUTES(object)
end
end
result $\leftarrow$ object $\cup$ objectAttributes;
if result $\neq$ failure then
return result
else
return failure
end
```

Semantic Similarities in Triplets

Using the similarity between triplets, we can easily determine which sentences are talking or describing the same factoid and hence replacing that factoid by a sentence which represents the entirety of said fact, we can eliminate the redundancy generated by different articles. After the sentences have been broken down into triplets which are then stemmed and lemmatized; we need to determine which sentences present similar information by determining the semantic similarity between the two; and this process is then done by using the Wu-Palmer Similarity heuristic [23]. As mentioned before, this is the step where we aim to eliminate redundant sentences, sentences with high semantic similarity occur in same clusters and are thus we can eliminate redundancy by picking only the most important sentence in said cluster. Wu-Palmer similarity is a Structure based similarity metric. Structure-based or edge counting measures represent the measures that use a function that computes the semantic similarity measure in ontology hierarchy structure (is-a, part-of). The function computes the length of the path linking the terms and on the po-sition of the terms in the taxonomy. Thus, the more similar two concepts are, the more links there are among the concepts and the more closely related they are [24].

Calculating Similarity Matrix

Similarity Matrix gives a sense of how similar elements is the matrix are to each other. That is for an element A and B in a matrix, the similarity score can be determined by checking row A and column B of the similarity matrix. These matrices play a pivotal role in the process of clustering similar elements together. To calculate the similarity matrix; we take all the triplets and semantic similarity is calculated between them by comparing them to all other triplets using the following formula.

CLUSTERING TRIPLETS

In the previous step; each sentence is mapped into a metric space whose dimensions are the related subject, predicate and object, and a relative distance metric: summarization problem can be thus formalized as the problem of determining in this metric space a set of triples, or a summary, that better represents the semantics of the document. This basically means we can now cluster the triplets into distinct non-redundant clusters based on the topics being spoken about in each cluster. For this purpose we use the OPTICS clustering algorithm.

OPTICS belongs to a group of Density based clustering algorithms. Density based clustering algorithms are those which locate regions of high density and separate them regions of low density. Here density means the total number of data points within a specified radius called denoted by in OPTICS. To understand OPTICS first we need to understand DBSCAN, which is the algorithm it has been inspired by.

Density Based Spatial Clustering of Applications with Noise(DBSCAN) is a data clustering algorithm. It was proposed by Martin Ester et. al. 1996 [25]. It is a density-based clustering algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).

DBSCAN takes two parameters: $\in$ and the minimum number of points required to form a dense region . It starts with an arbitrary starting point that has not been visited. This point's neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized environment of a different point and hence be made part of a cluster.

If a point is found to be a dense part of a cluster, its neighborhood is also part of that cluster. Hence, all points that are found within the neighborhood are added, as is their own neighborhood when they are also dense. This process continues until the density-connected cluster is completely found.

Algorithm 6: DBSCAN, $\in$

```
Data: D,
    Result: all points in a point P's ∈-neighbourhood foreach
Point P in Dataset D do
    if P is Visited then
    continue to next point
    end
    mark P as visited;
    if sizeof(NeighbourPts) <MinPts then
    mark P as NOISE
    else
    C = next cluster;
    expandCluster(P, NeighbourPts, C, , MinPts)
    end
    end
```

Algorithm 7: expandCluster

```
Data: P, NeighbourPts, C,∈ , MinPts
    add P to cluster C;
    for each point P' in NeighbourPts do
    if P' is not visited then
    mark P' as visited;
    NeighbourPts' = regionQuery(P',∈ );
    end
    if sizeof(NeighbourPts') MinPts then
    NeighbourPts = NeighbourPts joined with NeighbourPts' end
    if P' is not yet member of any cluster then add P' to cluster C
    end
    end
```

```
Algorithm 8: regionQuery
Data: P,∈
Result: all points in a point P's ∈-neighbourhood
```

SENTENCE SELECTION

The output from the previous sections is the clustering of all the sentences generated from the articles. Since the application generates an Extractive summary, the summary contains the important sentences collected from the articles AS-IS. So, once we are done clustering the sentences based on the information they provide and their semantic similarity, we are left with picking a single sentence from each cluster which clearly represents information given by said cluster. This process is done as follows:

1. Since OPTICS already determines the clustering order, the sentences are as per centroids of each clusters, thus they give the most amount of information as to what the cluster pertains to.
2. Arrange the centroid sentences in a chronological manner with respect to date of publishing. This ensures factual chronology.
3. Output all the selected and sorted sentences as a wholesome summary.

EVALUATION PLAN

We plan to evaluate our models on gold-standard datasets for the summarization task, such as DUC-2004. ROUGE is a recall-based metric which assesses how many n-grams in generated summaries appear in the human reference summaries. This metric is designed to evaluate extractive methods.

CONCLUSION

In this position paper we outlined our ongoing research on multidocument text summarization that presents a system to automatically collect, collate and summarize on-line news paper articles automatically based on a user submitted query. The dataset used for summarization is ad-hoc and is generated on-the-fly. Using preprocessing steps like Sentence tokenization, NER, Stemming and lemmatization and Triplet formation; the articles are broken down into manageable semantic atoms which are then clustered based on their semantic similarity.

REFERENCES

[1] B. Baldwin, R. Donaway, E. Hovy, E. Liddy, I. Mani, D. Marcu, K. McKeown, V. Mittal, M. Moens, D. Radev *et al.*, "An evaluation road map for summarization research," *TIDES, July*, 2000.

[2] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 1995, pp. 68–73.

[3] C.-Y. Lin and E. Hovy, "The automated acquisition of topic signatures for text summarization," in *Proceedings of the 18th conference on Computational linguistics-Volume 1.* Association for Computational Linguistics, 2000, pp. 495–501.

[4] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2001, pp. 406–407.

[5] M. Osborne, "Using maximum entropy for sentence extraction," in *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4.* Association for Computational Linguistics, 2002, pp. 1–8.

[6] K. M. Svore, L. Vanderwende, and C. J. Burges, "Using signals of human interest to enhance single-document summarization." in *AAAI*, 2008, pp. 1577–1580.

[7] ——, "Enhancing single-document summarization by combining ranknet and third-party sources." in *Emnlp-conll*, 2007, pp. 448–457.

[8] K. McKeown and D. R. Radev, "Generating summaries of multiple news articles," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 1995, pp. 74–82.

[9] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41,

1995.

[10] K. Ono, K. Sumita, and S. Miike, "Abstract generation based on rhetorical structure extraction," in *Proceedings of the 15th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1994, pp. 344–348.

[11] R. Barzilay and K. R. McKeown, "Sentence fusion for multidocument news summarization," *Computational Linguistics*, vol. 31, no. 3, pp. 297–328, 2005.

[12] A. Nenkova, L. Vanderwende, and K. McKeown, "A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 573–580.

[13] J. Bian, Y. Yang, and T.-S. Chua, "Multimedia summarization for trending topics in microblogs," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013, pp. 1807–1812.

[14] L. Hennig and D. Labor, "Topic-based multi-document summarization with probabilistic latent semantic analysis." in *Ranlp*, 2009, pp. 144–149.

[15] I. Mani and E. Bloedorn, "Multi-document summarization by graph search and matching," *arXiv preprint cmp-lg/9712004*, 1997.

[16] F. Amato, A. dâŁ™Acierno, F. Colace, V. Moscato, A. Penta, and A. Picariello, "Semantic summarization of news from heterogeneous sources," in *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. Springer, 2016, pp. 305–314.

[17] M. Alruily, A. Ayesh, and H. Zedan, "Crime profiling for the arabic language using computational linguistic techniques," *Information Processing & Management*, vol. 50, no. 2, pp. 315–341, 2014.

[18] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006.

[19] J. Leskovec, M. Grobelnik, and N. Milic-Frayling, "Learning sub-structures of document semantic graphs for document summarization," 2004.

[20] S. Teufel and M. Moens, "Sentence extraction as a classification task," in *Proceedings of the ACL*, vol. 97, no. 1997, 1997, pp. 58–65.

[21] D. Brickley and R. V. Guha, "Rdf vocabulary description language 1.0: Rdf schema," 2004.

[22] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The penn treebank: annotating predicate argument structure," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 114–119.

[23] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1994, pp. 133–138.

[24] R. Richardson, A. F. Smeaton, and J. Murphy, "Using wordnet as a knowledge base for measuring semantic similarity between words," 1994.

[25] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.