Scientific Research Publishing

International Journal of Intelligence Science, 2017, 7, 55-71
http://www.scirp.org/journal/ijis
ISSN Online: 2163-0356
ISSN Print: 2163-0283

Extractive Summarization Using Structural Syntax, Term Expansion and Refinement

Mohamed Taybe Elhadi

Department of Computer Science, University of Zawia, Zawia, Libya
Email: mtelhadi@yahoo.com

How to cite this paper: Elhadi, M.T. (2017) Extractive Summarization Using Structural Syntax, Term Expansionand Refinement. *International Journal of Intelligence Science*, **7**, 55-71.
https://doi.org/10.4236/ijis.2017.73004

Received: June 26, 2017
Accepted: July 28, 2017
Published: July 31, 2017

Abstract

This paper investigates a procedure developed and reports on experiments performed to studying the utility of applying a combined structural property of a text's sentences and term expansion using WordNet [1] and a local thesaurus [2] in the selection of the most appropriate extractive text summarization for a particular document. Sentences were tagged and normalized then subjected to the Longest Common Subsequence (LCS) algorithm [3] [4] for the selection of the most similar subset of sentences. Calculated similarity was based on LCS of pairs of sentences that make up the document. A normalized score was calculated and used to rank sentences. A selected top subset of the most similar sentences was then tokenized to produce a set of important keywords or terms. The produced terms were further expanded into two subsets using 1) WorldNet; and 2) a local electronic dictionary/thesaurus. The three sets obtained (the original and the expanded two) were then re-cycled to further refine and expand the list of selected sentences from the original document. The process was repeated a number of times in order to find the best representative set of sentences. A final set of the top (best) sentences was selected as candidate sentences for summarization. In order to verify the utility of the procedure, a number of experiments were conducted using an email corpus. The results were compared to those produced by human annotators as well as to results produced using some basic sentences similarity calculation method. Produced results were very encouraging and compared well to those of human annotators and Jacquard sentences similarity.

Keywords

Data Extractive Summarization, Syntactical Structures, Sentence Similarity, Longest Common Subsequence, Term Expansion, WordNet, Local Thesaurus

1. Introduction

The growth of the web and the emergence of digital libraries make text analysis and similarity calculations an important technique for many applications. The multiple lingualism of such data explosion further necessitates the need for more robust, efficient and generalized tools and techniques to facilitate the utilization of available content. Text representations and processing have become an important backbone of many tools and applications including text mining [5] [6], summarization [7]-[17], clustering [18], categorization [19] [20], copy-detection [21] [22] [23], plagiarism [24] [25], web-search [26] [27], information retrieval [28] and computational biology [29] [30] [31].

This paper reports on work conducted to investigate the use of syntactical structures, namely POS-tagging of English sentences in the selection of parts of a document to be used as candidates for extractive summarization. The procedure uses POS tagging [32] [33] and LCS [3] [4] combined with term expansion using WordNet [1] and a local thesaurus [2] in the selection of the most appropriate extractive text summarization for a document. The produced sentences (extractive summary) of the document were the results of calculations based on the use of a set of selected common subsequences that was the result of the POS-tagged document' sentences. The results were further refined using term sets that were expanded into two subsets using WorldNet and a local electronic dictionary/thesaurus.

At first syntactical features of the sentences within the text were represented as POS-tags using TreeTagger [32] [33]. After that each document's tagged strings were further compared using LCS for common syntactical structures. A normalized score between 0 and 1 was calculated for each pair of sentences using the longest common subsequences to produce a final measure of similarity. An initial set of sentences was produced. The produced sentences were selected for being the topmost similar based on a predefined cut of value or mere selection of top-n sentences.

Further processing of the initial candidate set of sentences was performed, where a set of terms was produced from the set of candidate sentences to produce an initial set of terms or keywords (restricted to verbs, nouns and or adjectives and adverbs). The new set of terms was then used to expand the set of candidate sentences with any sentence that shares same terms in the original document. The expanded set of candidate sentences was further subjected to the same process again. The initial set of terms can either be used as is, or further improved using some global sources such as WordNet or a local resource such as a thesaurus or both. These experiments have used the initial set of terms as is, extended with WordNet and local thesaurus. This cycle can be performed any number of times to produce more refined sets of candidate sentences.

As an experimental validation of the adopted procedure, a dataset made up of real emails along with a human annotation of important sentences was used [5]. Obtained results were also compared to those that can be produced using sentence-based Jacquard coefficient similarity [3]. Results obtained have showed the utility of the approach in generating a set of candidate sentences that can be used for extractive summarizations and other similarity-based work.

All in all, a number of important processing tasks were performed these experiments including text tagging and basic text preprocessing. This aimed to make a reduction of a document' sentences into a set of POS tags without exclusion of any stop words, stemming or removal of numbers, punctuation or special characters. Each tag of each produced string was replaced by a single character. Mapping of similar tags such as verbs, nouns, adjectives and adverbs into single character or symbol can be applied to further reduce the size of the produced string to better improve efficiency of LCS processing.

Each tagged string of the original sentence was fed into LCS module to find the length of the most common-subsequence. Pairs of strings were then compared and scored based on a normalized value of the length of the longest common subsequence.

The most similar sentences were further analyzed to find set of words (terms) to be used for fetching of related sentences from the original document. Before using the collection of new sentences, the produced set of terms was expand-on using WordNet or a local dictionary. Top k-sentences were selected as candidate subset of sentences that can be used for extractive summarization.

The rest of the paper is made up of Section 2 on related work; Section 3 on the proposed procedure; Section 4 on the experiments conducted, and the document collections used; Section 5 on results analysis and Section 6 on conclusions and future work.

2. Related Work

Multi-target text summarization by humans involves full understanding, interpretation and generation of an abstract of documents. Such a task is not easy for the average person, talk less of a computer program. It is a very critical human cognitive activity whose objective is to sum up the main points of a long text. Automatic text summarization, the automation of this critical human activity is normally considered part of machine learning and data mining fields. It is typically utilized in a variety of fields such as search engines, document summarizations, and other non-typical fields such as image collections and videos. Automatic summarization involves methods and techniques from a variety of related fields that share text analysis and processing.

Tasks of text processing and data analysis have become a necessity in this ever-expanding field of text analysis and processing. Work on automatic text summarization [15] [16] [34]-[43] aims to make it easier and more efficient to create applications related to Natural Language Processing, such as Information Retrieval, Question Answering or Text Comprehension.

Automatic text summarization can be defined as the process of reducing a given text using a computer program to create a set of important points that can be extracted from the original document. Many tools and technologies with related algorithms have been developed and deployed to make a coherent summary of documents. Such methods take into account length, writing style and syntax using machine learning and other techniques [44]. All such tools share the major objective of creating a set (or subset) from the original document that works as a representative summary or abstract of the entire document. Summarization techniques and algorithms try to find subsets of objects which cover informational content of a single document or a group of documents.

Automatic summarization techniques can also be categorized based on the number of documents involved (single document versus multi-documents), the genre where a generic summarization which creates a generic summary of the documents versus query relevant summarization which creates a summary that selects objects from the original document that are relevant to some specific query. Query-focused summaries enable users to find more relevant documents more accurately, with less need to consult the full text of the document [17].

Most commonly, however, automatic summarization is categorized based on the type of produced summary which can be extractive or abstractive. In extractive [10] [42] summarization the summary is created by reusing portions (words, sentences … etc.) of the input text. As for abstractive [11] [12] [13] [14] summarization a summary is created by regenerating the extracted content from an internal representations of important concepts present in the document.

Extractive summarization works by selecting parts of the words, phrases, or sentences in the original text to form the summary while abstractive summarization builds an internal representation based on semantics used by natural language generation methods to create the summary.

Most researchers have looked at summarization as mere extraction of terms, phrases or sentences. They focus on extractive methods due in part to the difficulty of producing semantically generated summaries. The emphases of automatic extractive summarization is on the important issue of how can a system find and decide which sentences are important.

Lehn's work [11], considered one of the earliest attempts at automatic summarization, suggested a basic idea where sentences that convey important contents are those that contain some content descriptive words. Most of his work was based on finding the extracts from a given text depending on manually generated rules using sentence position, word formatting, word frequency and others clues [34] [45] [46]. The problem with this view is in its dependency on the format and position in the text rather than the semantics of text. Other early summarization systems such as FRUMP, SUMMONS, CIRCUS and SUMMARIST [47] [48] were based on the use of pre-defined patterns that are labor intensive. Patterns would trigger certain templates to be filled as the text is read [49]. Other techniques and algorithms which naturally model summarization problems were TextRank, Page Rank, Sub-modular set function, and Maximal Marginal Relevance (MMR) [50] [51] [52] [53] [54]. In many research work, cue words, title words, and sentence location for determining the sentence weights were used [44] [45] [55] [56].

Some researchers have represented a document as an undirected graph with nodes representing the sentences [10] [57] [58] [59]. These nodes in the graph that are connected were thus a representation of relatedness characterized by the value of the cosine similarity of their corresponding sentences. Sentences which were more similar to other sentences in the document are considered important and were included in the extractive summary.

Semantic graph based techniques [16] extract Subject-Object-Predicate triplets from the sentences that were then used to generate a graph of the document. Machine learning techniques are used to select a subpart of the graph where the sentences in the sub-part would make up the summary. Naïve Bays, Neural Networks and Hidden Markov Model (HMM) [12] were some of the machine learning methods used in summarization.

Testing and evaluation of summarization systems is a critical aspect that has been performed using all types of data sets and corpuses [8] [60] [61] [62]. Emails present one change in which such systems can be tested and verified. One of the first attempts that uses extraction of important phrases from emails as a way of email summarization is in [63] [64]. In [63], researchers focused on thread summarization using content and structural features to group sentences as "relevant" and "not relevant". Other researchers used a scoring-based summarization to generate "thread overviews" on mailing lists. In particular, [65] assumed that topical consistency can be maintained by selecting sentences with higher POS overlap with the root message. They based sentence score on POS overlap with the subject line and the root message. Whereas, [64] looked at thread summary creation more like an online group decision-making process using structure and Singular Value Decomposition (SVD) [66] on words bags to calculate a unique sentence scoring.

Using a supervised classifier, and a linguistically driven post-process to mark sentences as task descriptions, the SmartMail [67] was created to identify "action items" in a message by providing a task-focused summary consisting of a list of action items. A large email corpus was constructed representing each sentence on a large set of features with SVM classifiers trained to identify "task" sentences which, in turn, were utilized to obtain logical forms and task descriptions.

The idea of summarizing email threads using multi-candidate reduction as a framework [67] for abstractive multi-document summarization was used in [68]. They filtered sentences and compressed them in two ways in which they refer to a "parse-and-trim", and a Hidden Markov Model approach.

Ranked sentences using clue words through the construction of a Fragment Quotation Graph to capture the flow of a conversation in a thread was developed and used in [69]. A score for each sentence is assigned using the graph based on a test corpus that was built from 20 different Enron threads. The authors' approach outperformed MEAD and RIPPER–on this test set [69].

In [70] the authors presented a transformation for summarizing emails using an ontology that was populated by entities and relationships present in the email. The ontology could be learned very accurately with classifiers trained on a large set of features. It was then used to generate a summary maximizing an objective function relating sentence and entity weights. Work on extending the problem of keyword extraction in a supervised setting using a decision tree and a genetic-algorithm-based classifier to classify phrases in a document as key phrase or not was presented in [71].

One of the main tasks found in summarization as well as other text processing work had to do with evaluation of relatedness or similarity of parts of text, be it words, sentences or larger portions including whole documents. Different methods and approaches have been used to tackle this issue of similarities between documents using semantically, syntactical or semantic features. Semantic similarity received less attention for the inherent difficulties of representing semantics and the limitations on assessment coverage of user studies [54] [72]. Commonly used methods for determination of similarity include fingerprinting [21], Information Retrieval [28] and other hybrid techniques [24] [44]. In Information Retrieval models, more emphasis was put on representing documents by their words and word frequencies. Indexing with an appropriate model to evaluate similarities between documents was also used.

The combined use of syntactical POS tagging and text processing methods for the purpose of text similarity calculations and its applications was used in this recent work [72]-[77]. It was based on the intuition that similar (exact) documents would have similar (exact) syntactical structures. Documents that contain reused portions of other documents or are written by the same author or on the same topic would contain similar structures.

Looking at a lump of text as a string made of meaningful, well defined and numerable units (alphabets), means that a modified (and similar) text can be thought of as an intervention or application of edit operations commonly mentioned in bio-sequences analysis of insertions, deletions and substitutions.

3. Proposed Procedure

A brief description of the proposed procedure is shown in **Figure 1**.

Steps of the used procedure are briefly described next.

3.1. Text Tagging and Pre-Processing

This step makes a reduction of a document' sentences into a set of POS tags without exclusion of any stop words, stemming or removal of numbers, punctuation or special characters. Since, LCS algorithm handles characters, each tag of each produced string has been replaced by a single character. More simplification and reduction can be obtained through the mapping of similar tags such as verbs, nouns, adjectives and adverbs into single characters or symbols. This reduction can produce shorter strings, which is better for LCS calculation efficiency.

3.2. LCS-Processing

Each sentence' string of tags was then fed into an LCS module to produce the

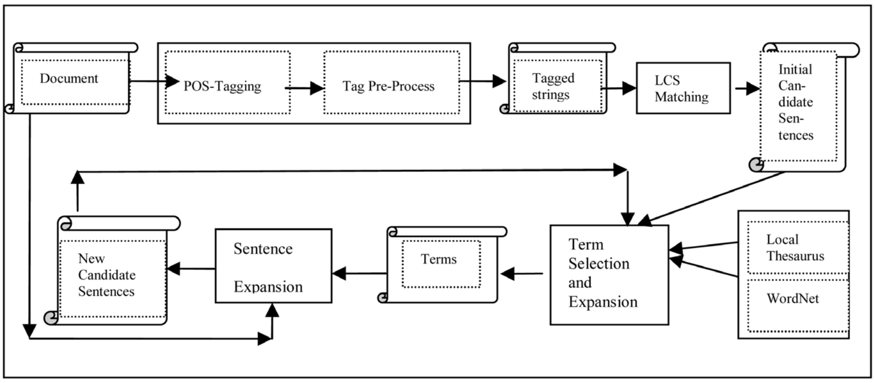


Figure 1. Overall depiction of the proposed procedure.

length of the most common-subsequence. Pairs of sentences were then compared and scored based on a normalized value of the length of the longest common subsequence and the tagged strings.

3.3. Tokenization, Term Selection and Expansion

The most similar sentences (based on the normalized LCS score) were then further analyzed to produce a set of terms (bag of words) to be used as keywords for collection of related sentences from the original document. All sentences that shared any of the key words were collected to be used for the next stage.

Before terms were used for collection of new sentences, the produced set of terms were subjected into a module that further expanded the set of keywords using either 1) WordNet; or 2) an electronic thesaurus-dictionary.

3.4. Subset of Candidate Sentence Selection

Once the procedure was applied a sufficient number of times, top sentences were selected as candidate subset of sentences that can be used for extractive summarization. The set is considered as a top k-sentences or any set of sentences that lay above a certain threshold value.

4. Dataset and Experiments

To evaluate the proposed procedure, it was applied on a subset of emails that were taken from [5] collection. The original email dataset consisted of a set of emails that were manually annotated with summaries and keywords and contained both single and thread emails. It totaled 349 annotated emails and threads. The dataset was developed for use by automatic summarization methods and other extraction experiments.

According to the developers, 319 emails of the 349 that were annotated came from the Enron corpus [12]. Thirty other emails were provided by volunteers. The set consists of a total of more than 100,000 words and close to 7000 sentences. The emails were classified as either corporate which refers to any communication within work environment; or private which refer to two different sets of private emails, the first was taken from the Enron collection and the second was mainly provided by volunteers from their own private mailboxes.

As per the developers of the email corpus, emails were manually annotated by two independent annotators generating 1) an abstractive summary; 2) a set of important sentences (extractive summary); 3) a set of key-phrases; and 4) a classification of the emails as either corporate or private.

For the purpose of this work, it was enough to use a subset of the corpus. A private single email collection (referred to as PSS) was used. The subset was made of 103 private emails along with two sets of sentences that were provided by human annotators. That gave a total of 206 extractive summaries.

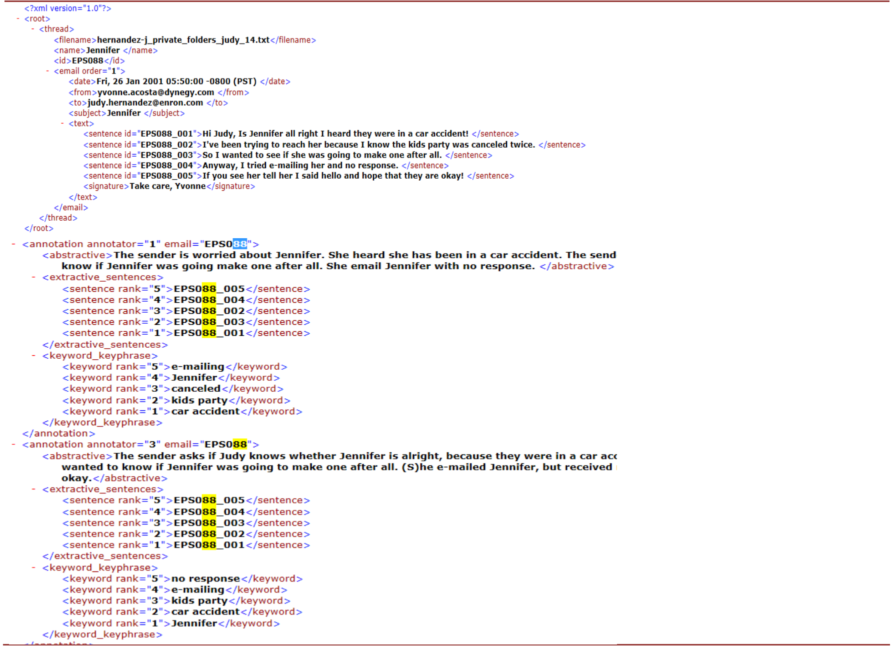
As can be seen from the samples provided in Table 1, the original email corpus was formatted using XML. The set of private single emails texts were extracted for each email in the test corpus along with their respective extractive summaries. The summary is made of 5 sentences suggested by the two human annotators. The annotators were identified as 1 and 3 in the original corpus, thus, the two sets PSS-A1 and PSS-A3 were created to correspond to the two annotators respectively. The two human annotators produced two sets that were not identical as expected.

Table 2 is a sample that shows the same email (088) along with produced POS Tags (for one sentence only) as well as the final and the much reduced string and terms.

All in all, the following comparisons were performed on the results:

1) Comparison of the used procedure produced results (sub set of sentences) obtained against the PSS-A1 set to its provided human annotated sentences.

Table 1. Sample of email number 0088 and part of its annotations and human made summaries.



```
  <?xml version="1.0"?>
- <root>
  - <thread>
      <filename>hernandez-j_private_folders_judy_14.txt</filename>
      <name>Jennifer </name>
      <id>EPS088</id>
    - <email order="1">
        <date>Fri, 26 Jan 2001 05:50:00 -0800 (PST) </date>
        <from>yvonne.acosta@dynegy.com </from>
        <to>judy.hernandez@enron.com </to>
        <subject>Jennifer </subject>
      - <text>
          <sentence id="EPS088_001">Hi Judy, Is Jennifer all right I heard they were in a car accident! </sentence>
          <sentence id="EPS088_002">I've been trying to reach her because I know the kids party was canceled twice. </sentence>
          <sentence id="EPS088_003">So I wanted to see if she was going to make one after all. </sentence>
          <sentence id="EPS088_004">Anyway, I tried e-mailing her and no response. </sentence>
          <sentence id="EPS088_005">If you see her tell her I said hello and hope that they are okay! </sentence>
          <signature>Take care, Yvonne</signature>
        </text>
      </email>
    </thread>
  </root>

- <annotation annotator="1" email="EPS088">
    <abstractive>The sender is worried about Jennifer. She heard she has been in a car accident. The send
      know if Jennifer was going make one after all. She email Jennifer with no response. </abstractive>
  - <extractive_sentences>
      <sentence rank="5">EPS088_005</sentence>
      <sentence rank="4">EPS088_004</sentence>
      <sentence rank="3">EPS088_002</sentence>
      <sentence rank="2">EPS088_003</sentence>
      <sentence rank="1">EPS088_001</sentence>
    </extractive_sentences>
  - <keyword_keyphrase>
      <keyword rank="5">e-mailing</keyword>
      <keyword rank="4">Jennifer</keyword>
      <keyword rank="3">canceled</keyword>
      <keyword rank="2">kids party</keyword>
      <keyword rank="1">car accident</keyword>
    </keyword_keyphrase>
  </annotation>
- <annotation annotator="3" email="EPS088">
    <abstractive>The sender asks if Judy knows whether Jennifer is alright, because they were in a car acc
      wanted to know if Jennifer was going to make one after all. (S)he e-mailed Jennifer, but received
      okay.</abstractive>
  - <extractive_sentences>
      <sentence rank="5">EPS088_005</sentence>
      <sentence rank="4">EPS088_004</sentence>
      <sentence rank="3">EPS088_003</sentence>
      <sentence rank="2">EPS088_002</sentence>
      <sentence rank="1">EPS088_001</sentence>
    </extractive_sentences>
  - <keyword_keyphrase>
      <keyword rank="5">no response</keyword>
      <keyword rank="4">e-mailing</keyword>
      <keyword rank="3">kids party</keyword>
      <keyword rank="2">car accident</keyword>
      <keyword rank="1">Jennifer</keyword>
    </keyword_keyphrase>
- </annotation>
```

Table 2. Sample of a very short email (text, tags, final tagged strings, and terms).

Email 0088 Text	Hi Judy, Is Jennifer all right I heard they were in a car accident! I've been trying to reach her because I know the kids party was canceled twice. So I wanted to see if she was going to make one after all. Anyway, I tried e-mailing her and no response. If you see her tell her I said hello and hope that they are okay!
Just the First Sentence POS Tags	Hi NP Hi Judy NP Judy , , , Is VBZ be Jennifer NP Jennifer all DT all right NN right
POS Final & Reduced Strings	jccffggpgBfiRincBffX ???++??++?*+?+?*?*??++ ioMqiciBGfnrjX ?***?*???*?++**. jiRMqcinMqacBX ?*?*??**?*??? jiRgiABfX ??*+???+ ciiqiiRfAviPDX ??*?*??*+?*??*-
Terms	sincerely#a maurc#n mohhlknaur#n Judy#n Jennifer#n right#n hear#v accident#n have#v reach#v know#v party#n cancel#v twice#a want#v make#v anyway#a response#n tell#v hello#n hope#v okay#a

2) Comparison of the procedure produced results (sub set of sentences) obtained against the PSS-A3 set to its provided human annotated sentences.

3) Correlation of the used procedure produced results to "how those of the two annotators compare to each other". That is we compared the annotators summaries to each other and then we correlated that to our results.

For the above 1) and 2) comparisons the top five sentences produced were compared to the 5 sentences produced by human annotators. Total match of results with a value of 1 meant that both sets contained the same sentences. Lesser values of (0.8, 0.6, 0.4, 0.2, 0) represented less of an agreement to no agreement at all. No regard was paid to the order of sentences in these experiments. All of the comparisons provided were performed using the produced sentences based on the following combinations.

4.1. Based on Original Terms (TT Set)

In this set the terms were selected from the candidate sentences as is without any expansion of the list of terms.

4.2. Based on the Expanded Terms Using WordNet Synonyms (ST Set)

In this set, the original terms set was expanded using synonyms from WordNet. In particular, nouns and verbs were used as seeds to expand the list using WordNet. WordNet [1] is a well known lexical database for English and other languages. It groups words into sets of synonyms called synsets. WordNet also provides short definitions, usage examples, and records a number of relations among the synonym sets or their members.

4.3. Based on the Expanded Terms Using a Local Dictionary (DT Set)

In this set the original terms set was expanded using synonyms from the Moby-saurus-thesaurus-dictionary [2]. The terms (nouns and verbs) were used to expand the original set using Mobysaurus. Mobysaurus is a free, feature-rich English thesaurus and dictionary. It integrates Moby Thesaurus II, Roget's Thesaurus, GCIDE Dictionary and WordNet.

In addition to the above methods, another important evaluation that was conducted was correlation of our results to those that can be obtained by mere comparison based-on words contained in the sentences using a standard Jacquard coefficient similarity [3]. Results are further discussed in the following section.

5. Results, Analysis and Discussions

In order to validate our procedure, a number of experiments were performed as already described above. The results of each of the performed steps are explained next.

Table 3 and **Table 4** show the comparison of the results produced by the suggested procedure when compared to each of the two human annotators. The numbers across the table represent percentage agreement between the produced abstracts and those of the annotators along with the average based on the whole 103 set of emails. The table shows the averages for each processing cycle.

The left column shows the type of term expansion method used (Thesaurus and WordNet) and the three cycles of refinements done. Cycle 1 represents the case with no expansion done, while the other cycles (2 and 3) represent the two sequential refinements for both Thesaurus and WordNet types.

The last three rows contain the maximum obtained value for the different results, the results of comparing the performance of the two human annotators (Annotator 1 vs. 3) and the results obtained using the Jacquard similarity coefficient (JSC Annotator 1 or 3).

Table 3. Results compared to those of the annotator 1 and those of annotator 1 vs. annotator 3.

No	Type	Avg	≥40	≥60	≥80	100
1	**Thesaurus Cycle 1**	36.5	56.31	31.1	10.68	0.97
2	**Thesaurus Cycle 2**	40.4	57.28	40.8	16.5	0.97
3	**Thesaurus Cycle 3**	42.1	62.14	41.8	15.53	0.97
4	**WordNet Cycle 1**	37.3	58.25	34	10.68	0.97
5	**WordNet Cycle 2**	41.7	61.17	40.8	16.5	0.97
6	**WordNet Cycle 3**	42.1	62.14	40.8	16.5	0.97
7	**MAX**	**42.1**	**62.14**	**41.8**	**16.5**	**0.97**
8	**Annotator 1 vs. 3**	63.3	85.44	72.8	43.69	11.7
9	**JSC Annotator-1**	40.1	59.87	36.3	15.21	2.27

Table 4. Results compared to those of the annotator 3 and those of annotator 1 vs. annotator 3.

No	Type	Avg.	≥40	≥60	≥80	100
1	Thesaurus Cycle 1	37.5	53.4	29.13	14.56	1.94
2	Thesaurus Cycle 2	43.5	67.96	35.92	21.36	3.88
3	Thesaurus Cycle 3	44.3	66.02	39.81	20.39	4.85
4	WordNet Cycle 1	37.3	53.4	28.16	15.53	1.94
5	WordNet Cycle 2	45.8	70.87	40.78	23.3	4.85
6	WordNet Cycle 3	46.2	70.87	43.69	22.33	3.88
7	**Maximum**	**46.2**	**70.87**	**43.69**	**23.3**	**4.85**
8	Annotator 1 vs. 3	63.3	85.44	72.82	43.69	11.65
9	JSC Annotator 3	41.5	65.05	33.98	15.86	3.24

5.1. Comparison of Results against Human Annotated Sentences (Set PSS-A1)

As is shown in Table 3, resulting abstracts obtained by our procedure were compared to those of the human annotator 1 and on three cycles using the thesaurus and WordNet synset expansion.

One noticeable thing is that in both cases of expansion using the thesaurus or WordNet higher averages were obtained in the expanded cycles of 1 and 2 than that of the base cycle of 1.

As is shown in Table 3, the best average obtained was 42.1 for the WordNet cycle 3 slightly better than that of the thesaurus. The 42.1 is still lower than that obtained when the two annotators were compared to each other. It is worth noting, as seen from the last row (JSC Annotator-1) versus (Maximum), that the obtained results outperform JSC in all cases except for the 100% case.

5.2. Comparison of Results against Human Annotated Sentences (Set PSS-A3)

As is show in Table 4, resulting abstracts obtained by the used procedure were compared to those of the human annotator 3 were slightly better than the case of human annotator 1. The results in both cases of expansion using a thesaurus or WordNet showed higher averages in the expanded cycles of 1 and 2 than that of the base cycle of 1.

The best average obtained, as shown in Table 4, was 46.2 for the WordNet cycle 3 was slightly better than that of the thesaurus. The 46.2 results were still lower than that obtained when the two annotators were compared to each other.

Interestingly, all the results were better than JSC across all columns and compared better than the case of annotator 1 compared to the case of annotator 1 vs. 3.

5.3. Correlation of Results vs. Two Annotators as They Compare to Each Other

As is shown in both tables (Table 3 and Table 4), when compared to each other, the annotators results showed variations. That is an indication of the difficulty and inconsistency of abstracting even when humans were involved.

The obtained results compared showed an under performance but reasonable results when compared with how the annotators compared to each other. Results showed that the procedure compares better with annotator 3 than 1.

5.4. Comparison of Results to Jacquard Similarity Coefficient (Mere Sentence Terms)

As is shown in Table 3 and Table 4, in both cases, results obtained outperformed the mere use of JSC on sentences. As a matter of fact, results were better in almost every case beyond cycle 1.

It can be seen that, a combined approach to extractive summarizations can perform reasonably well when compared to results obtained from human annotators. These experiments highlight the utility of combining structural (syntactical) features extracted as POS tags with semantically driven approach in both accelerating the processing that can be done using traditional string processing techniques such as LCS. It also highlights the functionality of combining such structural future with expanded keywords in improving the ranking and selection of important or representative sentences. The utility and functionality of such approach is further enhanced through the use of refinement cycles. Results are quite comparable to human annotators work and better than that of the mere use of common sentences comparison techniques such as Jacquard similarity coefficient.

6. Conclusions

A procedure for extractive summarization was developed and experiments were performed to investigate and validate the results. The procedure used an approach based on a combined POS tagging of sentences of a text document and term expansion using WordNet and a local thesaurus in the selection of the most appropriate extractive text summarization for that document. Sentences were POS-tagged and the produced strings were reduced into single character tags. Which were then subjected to Longest Common Subsequence (LCS) to calculate the similarity of the pairs of the sentences that make up the document producing a normalized score was obtained. A selected top subset of the most similar sentences was tokenized to produce a set of important keywords which were further expanded into two subsets using WorldNet and a local thesaurus. The two expanded sets obtained along with the original set of terms were re-cycled to further refine and expand the list of selected sentences from the original document. The process was repeated a number of times in order to find the best representative set of sentences. A final set of the top (best) sentences was selected as candidate sentences for summarization.

Experiments using an email corpus were conducted to verify the utility of the procedure. The obtained results were compared to those produced by human annotators on one hand and to those results produced using Jacquard similarity coefficient. Comparison and analysis of the obtained results using the developed procedure were very encouraging and compared reasonably to the human annotators and other methods. Since the approach does not require language-specific linguistic processing beyond identifying sentence and word boundaries, it can also be applied to other languages, for example. At the same time, incorporating syntactic and semantic information has led to superior results compared to plain similarity methods.

References

[1] Kilgarriff, A. (2000) Wordnet: An Electronic Lexical Database. MIT Press, Cambridge.

[2] http://www.mobysaurus.com

[3] Conesa, B.A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M.W., Gaffney, D.J., Elo, L.L., Zhang, X. and Mortazavi, A. (2016) A Survey of Best Practices for RNA-seq Data Analysis. *Genome Biology*, **17**, 13. https://doi.org/10.1186/s13059-016-0881-8

[4] Elhadi, M. and Al-Tobi, A. (2010) Refinements of Longest Common Subsequence Algorithm. 2010 *IEEE/ACS International Conference on In Computer Systems and Applications* (*AICCSA*), Washington DC, 16-19 May 2010, 1-5. https://doi.org/10.1109/AICCSA.2010.5586959

[5] Loza, V., Lahiri, S., Mihalcea, R. and Lai, P.H. (2014) Building a Dataset for Summarization and Keyword Extraction from Emails. InLREC, 2441-2446.

[6] Talib, R., Hanif, M.K., Ayesha, S. and Fatima, F. (2016) Text Mining: Techniques, Applications and Issues. *International Journal of Advanced Computer Science & Applications*, **1**, 414-418. https://doi.org/10.14569/IJACSA.2016.071153

[7] Pal, A.R., Maiti, P.K. and Saha, D. (2013) An Approach To Automatic Text Summarization Using Simplified Lesk Algorithm and Wordnet. *International Journal of Control Theory and Computer Modeling*, **3**, 15-23. https://doi.org/10.5121/ijctcm.2013.3502

[8] Hovy, E., Lin, C.Y., Zhou, L. and Fukumoto, J. (2006) Automated Summarization Evaluation with Basic Elements. *Proceedings of the Fifth Conference on Language Resources and Evaluation* (*LREC* 2006), Genoa, 22-28 May 2006, 604-611.

[9] André, P., Kittur, A. and Dow, S.P. (2014) Crowd Synthesis: Extracting Categories and Clusters from Complex Data. *Proceedings of the* 17*th ACM Conference on Computer Supported Cooperative Work & Social Computing*, Maryland, 15-19 February 2014, 989-998. https://doi.org/10.1145/2531602.2531653

[10] Mihalcea, R. (2004) Graph-Based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. *Proceedings of the* 2004 *ACL on Interactive Poster and Demonstration Sessions*, Barcelona, 21-26 July 2004.

[11] Luhn, H.P. (1958) The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, **2**, 159-165. https://doi.org/10.1147/rd.22.0159

[12] Aggarwal, C.C., Ed. (2014) Data Classification: Algorithms and Applications. CRC Press, Boca Raton, Florida.

[13] Klimt, B. and Yang, Y. (2004) The Enron Corpus: A New Dataset for Email Classification Research. In: Boulicaut, J.F., Esposito, F., Giannotti, F. and Pedreschi, D., Eds., *Machine Learning*: *ECML 2004*. *Lecture Notes in Computer Science*, Vol. 3201, Springer, Berlin, Heidelberg, 217-226.

[14] Ramshaw, L.A. and Marcus, M.P. (1999) Text Chunking Using Transformation-Based Learning. In: Armstrong, S., *et al.*, Eds., *Natural Language Processing Using Very Large Corpora*, Springer Netherlands, 157-176. https://doi.org/10.1007/978-94-017-2390-9_10

[15] Gupta, V. and Lehal, G.S. (2010) A Survey of Text Summarization Extractive Techniques. *Journal of Emerging Technologies in Web Intelligence*, **2**, 258-268. https://doi.org/10.4304/jetwi.2.3.258-268

[16] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B. and Kochut, K. (2017) Text Summarization Techniques: A Brief Survey. arXiv:1707.02268

[17] Nenkova, A. and McKeown, K. (2011) Automatic Summarization. *Foundations and Trends in Information Retrieval*, **5**, 103-233. https://doi.org/10.1561/1500000015

[18] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. and Bouras, A. (2014) A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. *IEEE Transactions on Emerging Topics in Computing*, **2**, 267-279. https://doi.org/10.1109/TETC.2014.2330519

[19] Aggarwal, C.C. and Reddy, C.K., Eds. (2013) Data Clustering: Algorithms and Applications. CRC Press, Boca Raton, Florida.

[20] Aggarwal, C.C. and Zhai, C. (2012) A Survey of Text Clustering Algorithms. In: *Mining Text Data*, Springer US, 77-128. https://doi.org/10.1007/978-1-4614-3223-4_4

[21] Yang, L. and Xi, J. (2015) Human Behavior Recognition: Semantics-Based Text Copy Detection Method. 2015 *First International Conference on Computational Intelligence Theory, Systems and Applications* (*CCITSA*), Yilan, 10-12 December 2015, 158-162. https://doi.org/10.1109/CCITSA.2015.28

[22] Elhadi, M. and Al-Tobi, A. (2010) Detection of Duplication in Documents and WebPages Based Documents Syntactical Structures through an Improved Longest Common Subsequence. *IJIPM*, **1**, 138-147. https://doi.org/10.4156/ijipm.vol1.issue1.16

[23] Potthast, M., Barrón-Cedeño, A., Stein, B. and Rosso, P. (2011) Cross-Language Plagiarism Detection. *Language Resources and Evaluation*, **45**, 45-62. https://doi.org/10.1007/s10579-009-9114-z

[24] Bin-Habtoor, A.S. and Zaher, M.A. (2012) A Survey on Plagiarism Detection Systems. *International Journal of Computer Theory and Engineering*, **4**, 185. https://doi.org/10.7763/IJCTE.2012.V4.447

[25] Osman, A.H., Salim, N. and Abuobieda, A. (2012) Survey of Text Plagiarism Detection. *Computer Engineering and Applications Journal* (*ComEngApp*), **1**, 37-45.

[26] Poinçot, P., Lesteven, S. and Murtagh, F. (1998) Comparison of Two “Document Similarity Search Engines. *ASP Conference Series*, **153**, 85.

[27] Elhadi, M. and Al-Tobi, A. (2009) Webpage Duplicate Detection Using Combined POS and Sequence Alignment Algorithm. 2009 *WRI World Congress on Computer Science and Information Engineering*, Los Angeles, 31 March-2 April 2009, 630-634. https://doi.org/10.1109/CSIE.2009.771

[28] Grossman, D.A. and Frieder, O. (2012) Information Retrieval: Algorithms and Heuristics. Springer Science & Business Media, Berlin.

[29] Pabinger, S., Dander, A., Fischer, M., Snajder, R., Sperk, M., Efremova, M., Krabichler, B., Speicher, M.R., Zschocke, J. and Trajanoski, Z. (2014) A Survey of Tools for Variant Analysis of Next-Generation Genome Sequencing Data. *Briefings in*

Bioinformatics, **15**, 256-278.

[30] Baral, C. (2004) Local Alignment: Smith-Waterman Algorithm, CSE 591: Computational Molecular Biology Course, Arizona State University.

[31] Li, H. and Homer, N. (2010) A Survey of Sequence Alignment Algorithms for Next-Generation Sequencing. *Briefings in Bioinformatics*, **11**, 473-483. https://doi.org/10.1093/bib/bbq015

[32] Schmid, H. (2013) Probabilistic Part-of-Speech Tagging Using Decision Trees. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf

[33] Schmid, H. (1995) TreeTagger—A Language Independent Part-of-Speech Tagger. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Vol. 43, 28.

[34] Dalianis, H. (2000) SweSum—A Text Summarizer for Swedish No. TRITA-NA-P0015. NADA, KTH, Stockholm.

[35] Hassel, M. (2007) Resource Lean and Portable Automatic Text Summarization [Internet]. http://www.csc.kth.se/~xmartin/papers/Nodalida03final.pdf

[36] Jones, K.S. (2007) Automatic Summarising: The State of the Art. *Information Processing & Management*, **43**, 1449-1481. https://doi.org/10.1016/j.ipm.2007.03.009

[37] Barzilay, R. and Elhadad, M. (1999) Using Lexical Chains for Text Summarization. In: Mani, I. and Maybury, M.T., Eds., *Advances in Automatic Text Summarization*, The MIT Press, Cambridge, 111-121.

[38] Hassel, M. (2003) Exploitation of Named Entities in Automatic Text Summarization for Swedish. 14*th Nordic Conference on Computational Linguistics*, Reykjavik, 30-31 May 2003.

[39] Nobata, C, Sekine, S., Isahara, H. and Grishman, R. (2002) Summarization System Integrated with Named Entity Tagging and IE Pattern Discovery. *Proceedings of the LREC*-2002 *Conference*, Canaria, May 2002, 1742-1745.

[40] Mani, I. and Maybury, M.T., Eds. (1999) Advances in Automatic Text Summarization. MIT Press, Cambridge.

[41] Mani, I. (2001) Automatic Summarization [Internet]. Vol. 3. John Benjamins Publishing, Amsterdam.

[42] Mani, I. (2001) Recent Developments in Text Summarization. *Proceedings of the Tenth International Conference on Information and Knowledge Management*, Atlanta, 5-10 October 2001, 529-531. https://doi.org/10.1145/502585.502677

[43] Dalianis, H. and Åström, E. (2001) SweNam—A Swedish Named Entity Recognizer. Technical Report, TRITANA-P0113, IPLab-189, KTH NADA.

[44] Bijalwan, V., Kumari, P., Pascual, J. and Semwal, V.B. (2014) Machine Learning Approach for Text and Document Mining. arXiv:1406.1580

[45] Edmundson, H.P. (1969) New Methods in Automatic Extracting. *Journal of the ACM* (*JACM*), **16**, 264-285. https://doi.org/10.1145/321510.321519

[46] Lin, C.Y. and Hovy, E. (1997) Identify Topics by Position. *Proceedings of the* 5*th Conference on Applied Natural Language Processing*, Washington DC, 31 March-3 April 1997, 283-290. https://doi.org/10.3115/974557.974599

[47] Hovy, E. and Lin, C.Y. (1998) Automated Text Summarization and the SUMMARIST System. *Proceedings of a Workshop*, Baltimore, 13-15 October 1998, 197-214.

[48] Chuang, W.T. and Yang, J. (2000) Extracting Sentence Segments for Text Summarization: A Machine Learning Approach. *Proceedings of the* 23*rd Annual International ACM SIGIR Conference on Research and Development in Information Re-*

trieval, Athens, 24 -28 July 2000, 152-159.

[49] Lin, C.Y. and Hovy, E. (2000) The Automated Acquisition of Topic Signatures for Text Summarization. *Proceedings of the* 18*th Conference on Computational Linguistics*, **1**, 495-501. https://doi.org/10.3115/990820.990892

[50] Mihalcea, R. and Tarau, P. (2004) TextRank: Bringing Order into Text. *EMNLP*, **4**, 404-411.

[51] Page, L., Brin, S., Motwani, R. and Winograd, T. (1999) The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford InfoLab, Stanford.

[52] Lin, H. and Bilmes, J. (2011) A Class of submodular Functions for Document Summarization. *Proceedings of the* 49*th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, **1**, 510-520.

[53] Murray, G., Renals, S. and Carletta, J. (2005) Extractive Summarization of Meeting Recordings. *Proceedings of the* 9*th European Conference on Speech Communication and Technology*, Lisbon, 4-8 September 2005, 593-596.

[54] Gambhir, M. and Gupta, V. (2017) Recent Automatic Text Summarization Techniques: A Survey. *Artificial Intelligence Review*, **47**, 1-66. https://doi.org/10.1007/s10462-016-9475-9

[55] Ajmal, E.B. and Haroon, R.P. (2016) Maximal Marginal Relevance Based Malayalam Text Summarization with Successive Thresholds. *International Journal on Cybernetics & Informatics*, **5**, 349-356. https://doi.org/10.5121/ijci.2016.5237

[56] Maguitman, A.G., Menczer, F., Roinestad, H. and Vespignani, A. (2005) Algorithmic Detection of Semantic Similarity. *Proceedings of the* 14*th International Conference on World Wide Web*, 10-14 May 2005, 107-116.

[57] Erkan, G. and Radev, D.R. (2004) Lexrank: Graph-Based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, **22**, 457-479.

[58] Litvak, M. and Last, M. (2008) Graph-Based Keyword Extraction for Single-Document Summarization. *Proceedings of the Workshop on Multi-Source Multilingual Information Extraction and Summarization*, Manchester, 23 August 2008, 17-24. https://doi.org/10.3115/1613172.1613178

[59] Palshikar, G.K. (2007) Keyword Extraction from a Single Document Using Centrality Measures. *International Conference on Pattern Recognition and Machine Intelligence*, Kolkata, 18 December 2007, 503-510. https://doi.org/10.1007/978-3-540-77046-6_62

[60] Lin, C.Y. (2004) Rouge: A Package for Automatic Evaluation of Summaries. *Workshop on Text Summarization Branches Out*, *Post-Conference Workshop of ACL*, Barcelona, 25 July 2004.

[61] Hassel, M. (2004) Evaluation of Automatic Text Summarizaiton: A Practical Implementation. Doctoral Dissertation, Numerisk analys och datalogi.

[62] Jing, H., Barzilay, R., McKeown, K. and Elhadad, M. (1998) Summarization Evaluation Methods: Experiments and Analysis. *AAAI Symposium on Intelligent Summarization*, Stanford, 23 March 1998, 51-59.

[63] Muresan, S., Tzoukermann, E. and Klavans, J.L. (2001) Combining Linguistic and Machine Learning Techniques for Email Summarization. *Proceedings of the* 2001 *Workshop on Computational Natural Language Learning*, **7**, 19. https://doi.org/10.3115/1117822.1117837

[64] Rambow, O., Shrestha, L., Chen, J. and Lauridsen, C. (2004) Summarizing Email Threads. *HLT-NAACL-Short* 04 *Proceedings of HLT-NAACL*, Boston, 2-7 May 2004, 105-108. https://doi.org/10.3115/1613984.1614011

[65] Nenkova, A. and Bagga, A. (2003) Email Classification for Contact Centers. *Proceedings of the* 2003 *ACM Symposium on Applied Computing*, Melbourne, FL, 9 March 2003, 789-792. https://doi.org/10.1145/952532.952689

[66] Shlens, J. (2014) A Tutorial on Principal Component Analysis. arXiv:1404.1100

[67] Zajic, D., Dorr, B.J., Lin, J. and Schwartz, R. (2007) Multi-Candidate Reduction: Sentence Compression as a Tool for Document Summarization Tasks. *Information Processing & Management*, **43**, 1549-1570. https://doi.org/10.1016/j.ipm.2007.01.016

[68] Corston-Oliver, S., Ringger, E., Gamon, M. and Campbell, R. (2004) Task-Focused Summarization of Email. ACL-04 Workshop: Text Summarization Branches Out, 43-50.

[69] Carenini, G., Ng, R.T. and Zhou, X. (2007) Summarizing Email Conversations with clue Words. *Proceedings of the* 16*th International Conference on World Wide Web*, Banff, 8 May 2007, 91-100. https://doi.org/10.1145/1242572.1242586

[70] Murray, G., Carenini, G. and Ng, R. (2010) Generating and Validating Abstracts of Meeting Conversations: A User Study. *Proceedings of the* 6*th International Natural Language Generation Conference*, Meath, 7 July 2010, 105-113.

[71] Carenini, G., Ng, R.T. and Zhou, X. (2008) Summarizing Emails with Conversational Cohesion and Subjectivity. *Proceedings of the* 46t*h Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, **8**, 353-361.

[72] Monostori, K., Finkel, R., Zaslavsky, A., Hodász, G. and Pataki, M. (2002) Comparison of Overlap Detection Techniques. *Computational Science—ICCS* 2002, Amsterdam, 21-24 April 2002, 51-60. https://doi.org/10.1007/3-540-46043-8_4

[73] Elhadi, M.T. (2012) Text Similarity Calculations Using Text and Syntactical Structures. 2012 7*th International Conference on Computing and Convergence Technology* (*ICCCT*), Seoul, 3 December 2012, 715-719.

[74] Elhadi, M. and Al-Tobi, A. (2009) Duplicate Detection in Documents and WebPages Using Improved Longest Common Subsequence and Documents Syntactical Structures. *Fourth International Conference on Computer Sciences and Convergence Information Technology*, Seoul, 24-26 November 2009, 679- 684. https://doi.org/10.1109/ICCIT.2009.235

[75] Elhadi, M. and Al-Tobi, A. (2008) Use of Text Syntactical Structures in Detection of Document Duplicates. *Third International Conference on Digital Information Management*, London, 13-16 November 2008, 520-525. https://doi.org/10.1109/ICDIM.2008.4746719

[76] Elhadi, M. and Al-Tobi, A. (2009) Part of Speech (POS) Tag Sets Reduction and Analysis Using Rough Set Techniques. *Rough Sets*, *Fuzzy Sets*, *Data Mining and Granular Computing*, Delhi, 15-18 December 2009, 223-230. https://doi.org/10.1007/978-3-642-10646-0_27

[77] Elhadi, M.T. (2016) Arabic Text Copy Detection Using Full, Reduced and Unique Syntactical Structures. *International Journal of Computer Applications*, **154**, 13-17. https://doi.org/10.5120/ijca2016912088