Using Frame-based Lexical Chains for Extracting Key Points from Texts

Sudabeh Mohamadi
University of Tehran
Tehran, Iran
E-mail: sumohamadi@ut.ac.ir

Kambiz Badie
Iran Telecom Research Center
Tehran, Iran
E-mail: k_badie@itrc.ac.ir

Ali Moeini
University of Tehran
Tehran, Iran
E-mail: moeini@ut.ac.ir

Abstract—**In last years, many automatic systems have been designed for text summarization and extracting key phrases, but still no system has been suggested for extracting key points. According to our definition, key points are high-level concepts extractable from a text that consist of words that may not necessarily exist in the original text. In this paper, we try to design an automatic system for extracting key points by using lexical chains. In this system, we use FrameNet for shallow semantic parsing of texts. As the final attempt, we present the set of tuples that contain important concepts of an original text with the related semantic roles. With use of generalization from parts onto whole, we can then have the claim of extracting a higher-level concept, which stands for a key point. Comparing the output of this system with human abstract, we perceived that 42 percent of cases generated by this system are similar to those generated by human.**

Keywords-automatic summarization; keyphrase extraction; abstract; lexical chain; generalization.

I. INTRODUCTION

Recently, the documents data are remarkably increasing and we need to have access to these data easily and rapidly. These data may belong to video, sound, text or image format. Text data may exist in web pages, books, articles, emails, documents of organizations, etc. Using these data leads to consuming much time to the extent that finding needed information becomes very hard and sometimes impossible. One of the ways for fast and suitable access to text information is automatic summarization of text [1]. The goal of automatic summarization is to take an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's need. In fact, summary is a brief statement that presents the main points in a concise form. There are two types of summary: extract and abstract.

In the former, summary is formed by reusing the portions of the main text like words or sentences. Words sequences that come into summary are the same as that of the main text. The words sequence can be used as phrases, sentences or paragraphs. In this type of summary, the most important information of the main text is copied to the final summary.

In an abstract, the content is an interpretation of an original text. In fact, abstracts consist of new phrases that describe the content of the original text. In this type of summary, words sequences that come into abstract are not the same as words sequence of the original text. Producing an abstract contains topic fusion and text generation stages. The main problem with text generation is that the new text should contain several sentences that must be coherent. For some specific applications, summary and abstract may be not suitable, as their structure of sentences may be complicated. For example, in search engines, we need to match key phrases or key points of texts with these of in the query [2]. In these cases, we can use key phrases or key points instead of summary. Key point and key phrase extraction is performed at word level while text summarization is performed at sentence level. Key phrases and key points can be considered as sets of words or concepts that present a brief representation of the original text. Key phrase extraction is highly related to automated text summarization where the most indicative words in a document are selected as key phrases.

In contrast with text summarization, key point extraction does not require coherence between sentences. In our definition, key points represent important concepts in the text that have the semantic relation with central topic of the original text. They consist of words that may not be necessarily in the original text. We can consider key points as a set of phrases that are semantically related to most of the portions of the text. In this paper, we try to extract key points.

Although information about text obtained from abstract are more than key points and key points cannot be considered as the alternatives for abstract but we can use them in specific applications such as indexing in search engines or text categorization. In addition, the key points

 ISBN: 978-1-61208-157-1

assigned to the text can help the reader distinguish whether a document is relevant or not.

As mentioned before, key points must have the most relevance with the concepts in the original text. The best way for presenting relevance between words and senses is using lexical chains. Lexical chains contain a set of words or senses related to each other with semantic relations. The words and senses in the same lexical chain are related to each other from the viewpoint of semantic relation. Portions of the text that are covered by each lexical chain are different from other lexical chains. Furthermore, the number of words and the type of relations between words would be different for each lexical chain [3]. Different criteria exist for measuring the strength of lexical chain. The number of words and the type of relations between them are particularly important here. We try extracting stronger chains because the strength of a chain indicates its importance [4].

There are several lexical resources for computing semantic distance between two nodes of lexical chains. For instance: Dictionaries, Thesauri, semantic nets, WordNet and FrameNet [5]. Most of the text summarization or key phrase extraction systems make use of WordNet ontology.

In this paper, for the first time, lexical chains are built by FrameNet ontology [6]. After the strength of chains was extracted, the extracted frames are generalized from parts to whole to obtain a high-level of concept. In the key phrase extraction or summarization systems, this stage does not exist. It is exclusive for our system. Two methods exist for this generalization. In one method, we generalize two sub-frames to a super-frame when both have the same super-frame. In second method, where the intermediate frame is super for the first frame and sub for other, we can generalize the first frame to the other frame. Finally, we obtain the list of the tuples that present the important concept of the original text with the related semantic roles. Output of this system can be used in clustering and classification properly.

The paper is organized as follows. In Section 2, we present the related work. The suggested approach includes five stages that would be explained in Section 3. We present the experimental results and the evaluation in Section 4. Finally, we conclude and suggest possible future improvements in Section 5.

II. OVERVIEW ON EXISTING APPROACHES

Currently, there is no system for key point extraction. However, many other technologies, such as Automatic Summarization [1], Information Retrieval [7] and keyword and key phrase Extraction [2] can be mentioned. In this section, we present a brief review on these technologies. The focus here is specifically on review of Automatic Key Phrase Extraction systems.

In 1999, Witten et al. [8] presented KEA algorithm for automatically extracting key phrases from text. KEA identifies candidate key phrases using lexical methods, calculates feature values for each candidate, and uses a machine-learning algorithm to predict which candidates can be suitable as key phrases. KEA's extraction algorithm has two stages: (1) Training that creates a model for identifying key phrases, using training documents where the author's key phrases are known. (2) Extraction that chooses key phrases from a new document. KEA finds less than half of the author's key phrases.

In 2000, Turney [2][9] used an approach for automatically extracting key phrases from texts as a supervised learning task. He performed two types of experiments to test his approach. His first set of experiments applied the C4.5 decision tree induction algorithm to this learning task and the second set of experiments applied the GenEx algorithm to the task. The experimental results showed that GenEx algorithm could generate better key phrases than C4.5 algorithm.

Avanzo and Magnini [10] presented the LAKE System (Learning Algorithm for Key phrase Extraction) that first considered a number of linguistic features to extract a list of candidate key phrases, then used a machine learning framework to select significant key phrases for a document. The two features that they used are reasonably effective but they did not consider any semantic features of key phrases. This system utilized key phrases extraction for summarization.

Turney and KEA algorithm used first occurrence position in text and frequency based features. Later, Hult [11] extended their systems by integrating more linguistic features like part of speech tags. He used four features: term frequency, collection frequency, relative position of the first occurrence and the POS tag(S) assigned to the term.

Hulth improved automatic keyword extraction, using more linguistic knowledge. He used supervised machine learning algorithm by adding linguistic knowledge to the representation such as syntactic features, rather than relying only on statistics such as term frequency and n-grams. He showed that keyword extraction from abstracts can be achieved by using simple statistical measures as well as syntactic information from the documents. He used approaches such as n-gram; chunking and pattern, then computed recall, precision and f-score for these approaches and then compared them. Extracting with chunking approach gives a better precision, while extracting all words or sequences of words matching any of a set of POS tag patterns gives a higher recall. The highest f-score is obtained by n-gram approach [11].

Ercan and Cicekli [12][13] are the first to use the lexical chains in keywords extraction. They proceeded to automatic keyword extraction of texts by supervised learning algorithm. They used lexical chains for this task and built them using the WordNet ontology. Ercan and Cicekli extracted keywords instead of key phrases because most of the phrases did not exist in WordNet data source. Thus, lexical chains were constructed just for words. They used seven features. Four of which are lexical chain's features. Then evaluated different combination of the seven features and concluded that lexical chain's features improves keyword extraction task. Their lexical chain based features focus on members of the lexical chains rather than the whole lexical chain.

In 2010, Sarkar et al. [14] presented a neural network based approach to key phrase extraction from scientific articles. For predicting whether a phrase is a key phrase or not, they used the estimated class probabilities as the confidence scores which are used in re-ranking the phrases belonging to a class: positive or negative and they finally compared their system with KEA and concluded that their proposed system performs better than KEA.

III. THE PROPOSED APPROACH

The suggested approach includes five stages that would be explained below.

A. Segmentation

In the first stage, after acquiring the input text, it must be segmented by a segmenter. The main reason of segmentation is to prevent from construction of huge chains. If lexical chains are constructed in the whole text, the size of chains becomes very large and consequently a large space is needed for their storage. On the other side, construction of chains in the entire text is very time consuming, because we must check the relation between each frame with the others for the whole text. Therefore, we divided the original text into smaller segments and then constructed chains in these segments.

One of the ways of text segmentation is to use text segmenters. The duty of the text segmenter is to divide the original text into segments that represent the same topic. It tries to break the text into thematically meaningful segments. There are several applications for text segmentation as text segmenter. One of these applications is Marphadorner. Marphadorner implements two linear segmentation methods, which use measures of lexical cohesion to produce segments: Marti Hearst's TextTiler [15] and Freddy Choi's [16]. Both try to find those portions of a text in which the vocabulary changes from one subtopic to another.

B. Shallow Semantic Parsing of Input Text

After original text was segmented, it must be parsed semantically. For this task, we use FrameNet dataset. In fact, in this stage, syntax and semantic structure of original text are identified. One of the applications for this goal is SHALMANESER. It is a SHALlow seMANtic parSER used to assign semantic classes –frames– and semantic roles –frame elements (FEs) – to original text automatically. To do so, it performs two stages. Firstly, disambiguates word senses that correspond to semantic classes with FRED and then assigns semantic roles by ROSY. The dataset for SHALMANESER is the FrameNet dataset [17].

C. Constructing Lexical Chains

Lexical chain construction is performed in three stages as follow:

1) Select Candidate Terms

Our goal is to extract the key points or the key concepts of the text, hence we consider frames as candidate terms that present concepts. The frame that is assigned to the lexical unit, expresses the concept of that lexical unit in the special position. So, the frame can be considered as the concept of its lexical unit, because when the word evokes a frame, it means that the frame is one of the word's concepts.

2) Select Appropriate Chain

We use FrameNet for recognition of the relation between frames and computing the semantic distance of frames as a lexical resource. In this algorithm, three types of relations are defined:

a) the extra-strong relation type: between a frame and its repetition occur.

b) the strong relation type: between two frames connected with one of frame-to-frame relation like these:

Inheritance- perspective on- sub frame- precedes- inchoative of- causative of- using [6]. You can also see details of these relations in FrameNet project. In the strong relation, two frames are connected directly.

c) the medium-strong relation type: between two frames connected to each other using another frame that is called intermediate.

In this algorithm, we just consider one frame as intermediate but to improve an extended algorithm, we can use relation with two or more intermediate frames.

3) Insert the Frame in the Chain

To select an appropriate chain, we added frames in order to place in the paragraph. Suppose n chains were constructed and now we want to find an appropriate chain for frame a. At first, we investigate the relation between a and each frame in chain j of n. If frame a has the extra-strong relation with one of the frames of j, a belongs to chain j. otherwise we must check strong relation like extra-strong relation for a. if strong relation was not found for it, we investigate medium-strong the same as other two relations.

According to this priority, we find the appropriate chains for the candidate frame. Three types of state can occur. If no appropriate chain is found, then a new chain is created and the candidate frame is inserted into it. Whenever, one appropriate chain was found, the frame is inserted into it. If two appropriate chains were defined, they are joined to each other. When the candidate frame is inserted in the chain, the chain will be updated. The new frame is then connected to the other frames in chain according to their types of relation with them.

Algorithm 1 is the pseudo-code describing lexical chains construction.

 ISBN: 978-1-61208-157-1

Algorithm 1: Lexical Chains Constructor (Frames)

```
-----------------------------------------------------------------------
Start
for each Frame a from (1…m) do
   for each Chain j from (1…n) do
     for each Frame(1…k)  in Chain(1…n) do
           if Frame_a has Extra-strong relation with Frame_k then
                    Add Frame_a to Chain_j
                    Update Chain_j
                    break
      else if Frame_a has strong relation with Frame_k then
                    Add Frame_a to Chain_j
                    Update Chain_j
                    break
      else if Frame_a has Medium-strong relation with Frame_k then
                    Add Frame_a to Chain_j
                    Update Chain_j
          end if
          else ConstructNewChain(Frame_a)
      end for
   end for
end for
end
-----------------------------------------------------------------------
```

D. *Semantic Distance Between Frames*

The semantic distance between frames depends on the relation type between them. We described three types of relations and according to them we must define three values for semantic distance.

Barzilay and Elhadad experimented different states and concluded that the following weights can be the best. So we use those in this approach [4].

In the first type, 10 should be added to the distance for each repetition. For example, if repetition of the frame is two in one paragraph , the distance becomes 10 and if it repeats n times in the paragraph, we must add " (n-1)*10" to the distance of chain. In the second type, where two frames are connected directly, the distance would be equal to 7. In the third type, when two frames are connected with other frame as intermediate, the distance would be 4.

E. *Scoring of Chains*

After the original text is converted to several chains by the presented algorithm, in this stage, we must identify the strongest chains for extraction. There is no formal way to evaluate chain strength, as there is no formal method to evaluate the quality of key points. Hence, we rely on an empirical methodology.

In our approach, we select the number of texts. The text has been parsed by using FrameNet dataset. In the beginning, we construct chains for those described above, and then we score chains according to different criterion. There are several formal measures on the chain for scoring as follows: chain length, number of chain's member and weight of relations between members of chains. Elhadad and Barzilay have presented other criteria like: distribution in the text, text span covered by the chain, density, graph topology and number of repetition. They concluded that only the length is a good predictor of the strength of a chain. They supposed that the length is the number of occurrences of members of the chain that we call number of chain's member [3][4].

In our algorithm, we use four different features for scoring the chains

Feature 1: the number of chain members

In this feature, we compute the number of chain's member where score of each chain is equal to this.

Feature 2: sum of the weight of frame-to-frame relations

In this feature, score of chain is equal to sum of the weight of relations. The way of scoring was described previously. Notice that whenever there is more than one type of relation between two frames, only the relation with more weight is considered.

Feature 3: Feature 1+Feature 2

This feature is created by sum of two former features. Sometimes, the number of frames is high, while the semantic distance between them is weak. This feature balances them.

Feature 4: score (chain) = length * homogeneity

Barzilay and Elhadad experimented different features and concluded that this feature is the best for extracting keywords. In this formula

(1) Homogeneity =1- (number of distinct occurrence / length).

IV. EXPERIMENTAL RESULTS AND EVALUATION

It should be mentioned in the beginning of this section that since no system still exists for extracting key points in the way we have elaborated in this paper, our basis for comparison is just human being who is asked to extract the key points from texts in an intuitive manner.

A. *Extracting the Strong Chains*

In this stage, we must extract chains with maximum score. For this goal, we need to have one threshold. A selected threshold for this algorithm is: average (scores) +2 * standard deviation (scores) the same as Brazilay and Elhadad criterion. So we recognize chains with scores higher than threshold as strong chains and extract them for use in key points. In fact, the chain would be extracted if

(2) Score (chain)>average (scores) + 2*standard deviation.

B. *Generalization From Parts to Whole*

In this stage, we perform generalization from parts to whole. If two frames have the medium strong relation, that means they are connected to each other with an intermediate frame, and we can generalize them in two ways. In this state, we achieve a higher level of concepts. For example

(1) Alice writes a note with pen.

(2) Alice draws a plan with pencil.

In (1), author writes with pen tool and in (2) creator draws with pencil tool. Write evokes Text_creation frame and draw evokes Create_physical_artwork. Relation of them is the same as Figure 1.

The Intentionally_create frame is the intermediate frame for Text_creation and Create_physical_artwork frames. This frame is not seen in the original text but it is a super-frame for the other two and both of them can be generalized to Intentionally_create. This frame has two core frame elements: creator and created_entity. In these sentences, creator is assigned to Alice and creator_entity is assigned to note and plan. "instrument" is one of the non_core elements for the Intentionally_create that has been evoked by pen and pencil. As a consequence, after generalization, the following tuples are created.

(Intentionally_create, Creator, Creator_entity, instrument)
(Intentionally_create, Alice, Note/plan, pen/pencil).

In this example, the intermediate frame is super-frame for both frames, so both frames are generalized to this frame. If intermediate frame is super-frame for one of the frames and sub-frame for the other, we can generalize sub-frame of intermediate to super-frame of it. In Figure 2, frame 2 is the intermediate for 3 and 1. Therefore, frame 1 is generalized to 3.

This stage is the final stage for key point extraction systems. After this, we achieve the list of tuples that contains a frame as the first member and frame elements as the other members. These tuples indicate the main concepts in original texts.

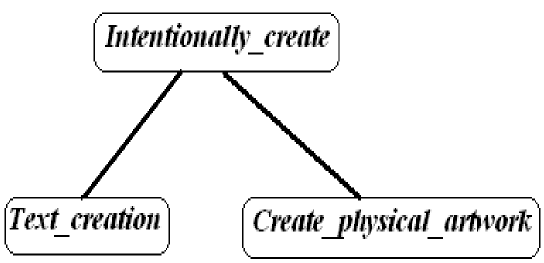

Figure 1. Example of relation between two frame

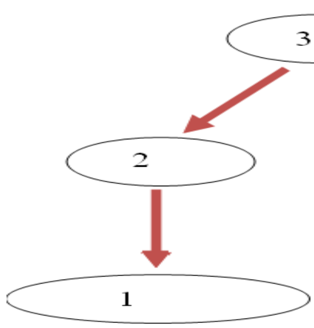

Figure 2. Generalization from parts to whole

C. Evaluation

For evaluation of this system, we use the texts that exist in FrameNet project of Berkeley [6]. FrameNet contains a number of full texts with their annotation and FE and frames would be included in them. We choose five of them to experiment our algorithm. Our system extracts key points from these, and then compares the output of automatic system with human extraction key points. Five students are chosen for extracting key points from the full texts. Two texts are given to each of them so each text is investigated by two students. We can use three performance criteria to evaluate this system [2].

(3) Recall = correct / (correct+missed)

(4) Precision = correct / (correct+wrong)

(5) F-measure = (2*recall*precision) / (recall+precision)

As it is seen in Table 1 to Table 3, Feature 1, Feature 2 and Feature 3 are very similar to each other and Feature 4 gives the worst result. The recall of Feature1 is better than the two other features but with regard to precision and f-measure criteria, Feature 2 is better than other features and Feature 3 is better than Feature1.

Feature 4, which has its most emphasis on number of iteration of frames, gives poor results. This feature focuses on the concept frequency and ignores relations between the frames. Since our goal is extracting the key points, the relations between frames are very important. Therefore, the Feature 4 is not suitable.

In Table 4, the average of recall and precision is shown. Comparing these with f-measure, we conclude that the recall and precision are balanced because the average of them is very similar to f-measure.

Table 1. THE RECALL CRITERIA

	Feature 1	Feature 2	Feature 3	Feature 4
Text 1(Madonna)	52%	52%	50%	36%
Text 2(Stephanopoulos Crimes)	31%	28%	28%	25%
Text 3(Bell Ringing)	42%	42%	42%	42%
Text 4(Loma Prieta)	34%	34%	34%	16%
Text 5(Dublin)	62%	62%	60%	61%
Average	44%	43%	42%	36%

Table 2. THE PRECISION CRITERIA

	Feature 1	Feature 2	Feature 3	Feature 4
Text 1(Madonna)	44%	44%	44%	25%
Text 2(Stephanopoulos Crimes)	11%	14%	14%	6%
Text 3(Bell Ringing)	30%	30%	30%	30%
Text 4(Loma Prieta)	60%	60%	60%	58%
Text 5(Dublin)	67%	75%	73%	73%
Average	42%	44.6%	44.2%	38%

Table 3. THE F-MEASURE CRITERIA

	Feature 1	Feature 2	Feature 3	Feature 4
Text 1(Madonna)	47%	47%	47%	29%
Text 2(Stephanopoulos Crimes)	16%	18%	18%	35%
Text 3(Bell Ringing)	35%	35%	35%	35%
Text 4(Loma Prieta)	43%	43%	43%	22%
Text 5(Dublin)	64%	67%	65%	68%
Average	41%	42%	41%	37%

Table 4. THE AVERAGE OF RECALL AND PRECISION

	Feature 1	Feature 2	Feature 3	Feature 4
Text 1(Madonna)	48%	48%	47%	30%
Text 2(Stephanopoulos Crimes)	21%	21%	21%	15%
Text 3(Bell Ringing)	36%	36%	36%	36%
Text 4(Loma Prieta)	47%	47%	47%	37%
Text 5(Dublin)	63%	69%	66%	67%

In these formulae, the variable *correct* represents the number of times that the human-generated key phrase matches the machine-generated key phrase. The *wrong* variable represents the number of concepts that the machine extracts but the human does not. The *missed* variable represents the number of concepts, which are extracted by human but not by machine. These performance criteria have been brought in Table 1, 2 and 3.

V. CONCLUSION AND FUTURE WORK

There are some systems for summarization and extracting key phrases from text, but there is no system for key point extraction. In this paper, we presented an approach to the task of extracting key points and concepts from English texts. For this goal, we have used lexical chains that are constructed based on FrameNet ontology. We then experimented four features based on lexical chains and achieved the expected results. This system extracts key points as high-level concepts from the original text. In 42 percent of the cases, the concept which generated by this system is equal to the concept generated by human. Although the output is complicated and difficult to understand for usual users but this approach is very useful in classifying and clustering systems.

The suggested system depends much on semantic parsing systems. Therefore, the extension of our system would call for improvement of semantic parsing systems. One of the shortcomings of this work is that we only consider one intermediate frame for the third type of relation. In future, relation with more intermediate frames can also be considered. In addition, in this work, just the medium-strong relation generalizes from parts to whole. In the future, strong relation can be considered too. Also, we can investigate more features which are based on concepts in the chains instead of the whole chains.

REFERENCES

[1] V. Gupta and G.S. Lehal, "A Survey of Text Summarization Extractive Techniques" Journal of Emerging Technologies in Web Intelligence, Vol 2, pp. 258-268, Aug 2010.

[2] P.D. Turney, "Learning to extract keyphrases from text", National Research Council, Institute for Information Technology, Technical Report, 2002.

[3] C. Santos, "Alexia - acquisition of lexical chains for text summarization", Master's thesis, University of Beira Interior, 2006.

[4] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization", Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, pp. 111-121, 1997.

[5] G. Hirst and A. Budanitsky, "Lexical chains and semantic distance", Proc. Eurolan, 2001.

[6] J. Ruppenhofer, M. Ellsworth, M. Petruck, and C. Johnson, "FrameNet II: Extended theory and practice", International Computer Science Institute,University of California at Berkeley, 2006.

[7] E. Greengrass, "Information Retrieval: A Survey", Citeseer, DOD Technical Report TR-R52-008-001, November 2000.

[8] I.H. Witten, G.W. Paynter., E. Frank, and C. Gutwin, "KEA: Practical automatic keyphrase extraction", In Proc. of the 4th ACM Conference on Digital Libraries, pp. 254-256, 1999.

[9] P.D. Turney, "Learning algorithm for keyphrase extraction", Journal of Information Retrieval, pp. 303-336, 2000.

[10] E.D. Avanzo and B. Magnini , "Keyphrase extraction for summarization purposes: The LAKE system at DUC-2004", Proceedings of the Document Understanding Conference, Boston, USA., 2004.

[11] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge", In Proceedings of the 2003 Conference on Empirical Methods in Natural Language processing Japan, August 2003.

[12] G. Ercan and L. Cicekli, "Using lexical chains for keyword extraction", Information Processing & Management, pp. 1705-1714, 2007.

[13] G. Ercan, "Automated text summarization and keyphrase extraction", Master thesis, 2006.

[14] K. Sarkar, M. Nasipuri, and S. Ghose, "A new approach to keyphrase extraction using neural networks", Vol. 7, Issue 2, No 3, pp. 16-25, Arxiv preprint arXiv:1004.3274, 2010.

[15] M.A. Hearst, "TextTiling: Segmenting text into multi-paragraph subtopic passages", Computational linguistics, pp. 33-64, 1997.

[16] F.Y.Y. Choi, "Advances in domain independent linear text segmentation", Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, pp. 26-33, 2000.

[17] K. Erk and S. Pado, "SHALMANESER: a toolchain for Shallow Semantic Parsing", Proceedings of LREC, Genoa, Italy, 2006.