Knowledge based Automatic Summarization

Andrey Timofeyev and Ben Choi
Computer Science, Louisiana Tech University, Ruston, U.S.A.

Keywords: Automatic Summarization, Semantic Knowledge Base, Text Analysis, Knowledge Discovery, Natural Language Processing.

Abstract: This paper describes a knowledge based system for automatic summarization. The knowledge based system creates abstractive summary of texts by generalizing new concepts, detecting main topics, and composing new sentences. The knowledge based system is built on the Cyc development platform, which comprises the world's largest ontology of common sense knowledge and reasoning engine. The system is able to generate coherent and topically related new sentences by using syntactic structures and semantic features of the given documents, the knowledge base, and the reasoning engine. The system first performs knowledge acquisition by extracting syntactic structure of each sentence in the given documents, and by mapping the words and the relationships of words into Cyc knowledge base. Next, it performs knowledge discovery by using Cyc ontology and inference engine. New concepts are abstracted by exploring the ontology of the mapped concepts. Main topics are identified based on the clustering of the concepts. Then, the system performs knowledge representation for human readers by creating new English sentences to summarize the key concepts and the relationships of the concepts. The structures of the composed sentences extend beyond subject-predicate-object triplets by allowing adjective and adverb modifiers. The system was tested on various documents and webpages. The test results showed that the system is capable of creating new sentences that include generalized concepts not mentioned in the original text and is capable of combining information from different parts of the text to form a summary.

1 INTRODUCTION

In this paper, we propose a knowledge based system for automatic summarization by utilizing knowledge base and inference engine to provide semantic summarization. The system creates abstractive summary of the given documents. It is built on Cyc development platform that includes world's largest ontology of common sense knowledge and inference engine (Cycorp, 2017). The knowledge base and inference engine enable the system to abstract new concepts, not directly stated in the text. The system utilizes semantic features and syntactic structure of the text. In addition, the knowledge base provides domain knowledge about the subject matter and allows the system to exploit relations between concepts in the documents.

The proposed system is unsupervised and domain independent, only limited by the comprehensive ontology of the common sense knowledge provided by the knowledge base. It generalizes new abstract concepts based on the knowledge derived from the text. It automatically detects main topics described in the text. Moreover, it composes new English sentences for some of the most significant concepts. The created sentences form an abstractive summary, combining concepts from different parts of the input text.

Although vast majority of the research in automatic text summarization has been conducted by extractive methods, abstractive summarization is considered to be more desirable. Sophisticated abstractive method would require the ability to fuse information from different parts of the original text, to synthesize new information and to incorporate domain knowledge (Cheung & Penn, 2013). Our proposed system provides these abilities as well.

Our knowledge based system starts with knowledge acquisition by deriving syntactic structure of each sentence of the input text and by mapping words and their relations into Cyc knowledge base. Next, it performs knowledge discovery by generalizing concepts upward in the Cyc ontology and detecting main topics covered in the text. Then, it conducts knowledge representation by composing

Timofeyev A. and Choi B.
Knowledge based Automatic Summarization.
DOI: 10.5220/0006580303500356
In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR 2017)*, pages 350-356
ISBN: 978-989-758-271-4

new sentences for some of the most significant concepts defined in main topics. The structure of the created sentences consists of subject, predicate and object elements and their adjective and adverb modifiers, thus allowing the system to create new English sentences that have structure beyond simple subject-predicate-object triplets when available.

The system was implemented and tested on various documents and webpages. The results show that the system is able to detect main topics comprised in the text, identify key concepts defined in those topics and create new sentences that contain novel information not explicitly mentioned in the original text.

As an example, the sentence "*Big felis usually being natural predatory animal*" was automatically generated by the system resulting from analysing articles that describe different types of felines. Here, concept "*felis*" acts as a subject of the sentence, "*being*" is a predicate and "*predatory animal*" is an object. Subject "*felis*" was not mentioned in the text and was derived by knowledge discovery process. Each element has its modifier – adjective "*big*" for subject, adverb "*usually*" for predicate and adjective "*natural*" for object respectively. The modifiers were chosen by the system based on the analysis of occurrences of the concepts and relationships of the concepts.

The rest of the paper is organized as follows. Related research in automatic text summarization is outlined in Section 2. System workflow overview is provided in Section 3. Detailed description of summarization process is given in Sections 4, 5 and 6. Technical details of the implementation and description of the results are covered in Section 7. Conclusion and future research are discussed in Section 8.

2 RELATED RESEARCH

Automatic text summarization seeks to compose a concise and coherent version of the original text preserving the most important information. Computational community has studied automatic text summarization problem since late 1950s (Luhn, 1958). Studies in this area are generally divided into two main approaches – extractive and abstractive. Extractive text summarization aims to select the most important sentences from original text to form a summary. Such methods vary by different intermediate representations of the candidate sentences and different sentence scoring schemes (Nenkova & McKeown, 2012). Summaries created by extractive approach are highly relevant to the original text, but do not convey any new information. Most prominent methods in extractive text summarization use term frequency versus inverse document frequency (TF-IDF) metric (Hovy & Lin, 1998), (Radev, et al., 2004) and lexical chains for sentence representation (Barzilay & Elhadad, 1999), (Ye, et al., 2007). Statistical methods based on Latent Semantic Analysis (LSA), Bayesian topic modelling, Hidden Markov Model (HMM) and Conditional random field (CRF) derive underlying topics and use them as features for sentence selection (Gong & Liu, 2001), (Shen, et al., 2007). Graph methods tend to represent the text as a graph of connected concepts or sentences. Effectively traversing such graph representation helps to choose relevant sentences to form a summary (Mihalcea & Tarau, 2004), (Günes & Radev, 2004). Machine learning techniques are widely used to score candidate sentences. Such methods discover most informative sentences based on wide variety of features (Wong, et al., 2008), (Rodriguez & Laio, 2014). Despite significant advancements in the extractive text summarization, such approaches are not capable of semantic understanding and limited to the shallow knowledge contained in the text.

In contrast, abstractive text summarization aims to incorporate the meaning of the words and phrases and generalize knowledge not explicitly mentioned in the original text to form a summary. Phrase selection and merging methods in abstractive summarization aim to solve the problem of combining information from multiple sentences. Such methods construct clusters of phrases and then merge only informative ones to form summary sentences (Bing, et al., 2015). Graph transformation approaches convert original text into a form of sematic graph representation and then combine or reduce such representation with an aim of creating an abstractive summary. (Ganesan, et al., 2010), (Moawad & Aref, 2012). Summaries constructed by described methods consist of sentences not used in the original text, combining information from different parts, but such sentences do not convey new knowledge.

Several approaches attempt to incorporate semantic knowledge base into automatic text summarization by using WordNet lexical database (Barzilay & Elhadad, 1999), (Bellare, et al., 2004), (Pal & Saha, 2014). Major drawback of WordNet system is the lack of domain-specific and common sense knowledge. Unlike Cyc, WordNet does not have reasoning engine and natural language generation capabilities.

Our system is similar to one proposed in (Choi & Huang, 2010). In this work, the structure of created sentences has simple subject-predicate-object pattern and new sentences are only created for clusters of compatible sentences found in the original text.

Recent rapid development of deep learning contributes to automatic text summarization, improving state-of-the-art performance. Deep learning methods applied to both extractive (Nallapati, et al., 2017) and abstractive (Rush, et al., 2015) summarization show promising results, but such approaches require vast amount of training data and powerful computational resources.

Our abstractive text summarization system derives syntactic structure to combine information from different parts of the text, uses knowledge base to have background semantic knowledge and performs reasoning to abstract new concepts. To derive syntactic features, such as part of speech tags and dependency parser labels, system uses SpaCy – Python library of advanced natural language processing (Honnibal & Johnson, 2015). The system utilizes capabilities of world's largest ontology of common sense knowledge – Cyc (Cycorp, 2017). The knowledgebase provides semantic knowledge and inference engine.

3 SUMMARIZATION PROCESS OVERVIEW

Our proposed system consists of three main parts: knowledge acquisition, knowledge discovery and knowledge representation for human reader. The workflow of the system is outlined in Figure 1.

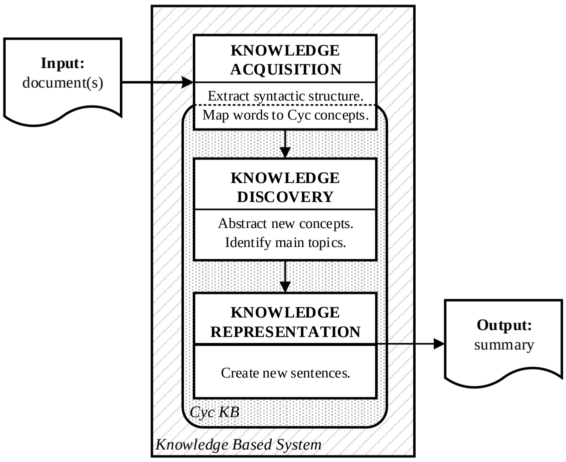

Figure 1: System workflow diagram.

During the knowledge acquisition process, our system takes documents as an input and transforms them into syntactic representation. Then, it maps each word in the text to the appropriate Cyc concept and assigns word's weight and associations to that concept. During the knowledge discovery process, the system finds ancestors for each mapped Cyc concept, records ancestor-descendant relation and adds scaled descendant weight and descendant associations to the ancestor concept. This process allows system to abstract new concepts not explicitly mentioned in the original text. Then, the system identifies main topics described in the text by clustering mapped Cyc concepts. During the knowledge representation process, the system creates English sentences for the most informative subjects identified in main topics. This process ensures that the summary sentences are composed using information from different parts of the text while preserving coherence to the main topics.

4 KNOWLEDGE ACQUISITION

Knowledge acquisition process consists of two sub-processes. The first sub-process – pre-processing, extracts syntactic structure of the given document. It separates text into sentences, lemmatizes each word and assigns part of speech tags and dependency parser associations. Then it counts the weights of the words and their associations. The second sub-process – mapping, finds matching Cyc concepts for each word in the input text. Once the system finds appropriate concept, it assigns word's weight and associations to that concept. Mapping sub-process is described in Figure 2.

Mapping sub-process:

- For each word obtained on preprocessing step:
  - Map word to corresponding Cyc concept;
  - Assign word's weight to mapped Cyc concept;
  - Map association head to corresponding Cyc concept;
  - Assign word's association to mapped Cyc concept.

Figure 2: Process of mapping words to Cyc concepts.

Word's weight is a frequency, the number of times it is mentioned in the text. The association is a relation between two words in a sentence, derived by the syntactic parser. Each association has a weight assigned to it that shows how many times two words were used together in the text. Higher weights represent stronger associations. Cyc ontology contains semantic knowledge about the concepts and our system enhances it with syntactic structure features. Semantic knowledge and syntactic structure are two crucial parts that make summary cohesive and meaningful.

5 KNOWLEDGE DISCOVERY

Knowledge discovery process consists of two sub-processes: (i) new concepts abstraction and (ii) main topics detection. The first sub-process starts by deriving ancestor for each concept mapped from the text. Then it assigns ancestor-descendant relation to derived ancestor and keeps track of descendants' scaled weight. The scaling is defined by generalization parameter α. Next, it adds descendants' weight and associations to ancestor concept if descendant-ratio is higher than the threshold. The threshold is defined by generalization parameter β. The descendant ratio is the number of mapped descendants divided by the number of all descendants of a concept. The parameters α and β regulate desired level of generalization. Higher α and lower β yield greater level of generalization giving more emphasis to ancestor terms. New concepts abstraction is an important part of summarization as it allows generalizing information derived from the input text. For example, our system can generalize "apple", "orange" and "mango" to an ancestor concept "fruit", which might not be mentioned in the text. Sub-process is described in Figure 3.

(i) New concepts abstraction sub-process:

- For each mapped Cyc concept:
  - Find concept's ancestor;
  - Record ancestor-descendant relation;
  - Update ancestor's number of descendants;
  - Update ancestor's descendants weight;
    - Scale descendant's weight by α.
- For each mapped Cyc concept that has descendants:
  - Find the number of concept's mapped descendants;
  - Find the number of all concept's descendants;
  - Calculate descendant ratio:

$$desc\_ratio = \frac{\#\ mapped\ descendants}{\#\ of\ all\ descendants}$$

  - If descendant-ratio is larger than β:
    - Add descendants' weight to ancestor's weight;
    - Add descendants' associations to ancestor associations;
      - Scale descendant's association weight by α.

Figure 3: Process of abstracting new concepts.

The second sub-process detects main topics in the text. The assumption is that the topics are represented by the most frequent micro theories in Cyc knowledge base. Micro theories are the clusters of concepts and facts typically representing one specific domain of knowledge. For example, #$MathMt is the micro theory containing mathematical knowledge. Micro theories are the basis of the knowledge representation in Cyc. Each concept begins to have a semantic meaning only in its defining micro theories (Matuszek, et al., 2006). To find the most frequent micro theories system derives defining micro theories for each mapped Cyc concept, counts frequencies of discovered micro theories and picks top-n micro theories with the highest frequencies. Sub-process is outlined in Figure 4.

(ii) Main topics detection sub-process:

- For each mapped Cyc concept:
  - Find defining micro theories;
- Count the frequencies of discovered micro theories;
- Pick top-n micro theories with highest frequencies.

Figure 4: Process of main topics detection.

6 KNOWLEDGE REPRESENTATION

Knowledge representation process starts by (i) choosing concepts with the highest subjectivity rank in each main topic detected by the knowledge discovery. These concepts become candidate subjects. Subjectivity rank is defined as the product of concept weight and subjectivity ratio. Subjectivity ratio is defined as the number of concept associations labelled as subject relation divided by the total number of concept associations. This ratio helps to identify concepts with the strongest subject roles in the text. Sub-process is described in Figure 5.

(i) Candidate subjects discovery sub-process:

- For each micro theory in top-n micro theories:
  - For each concept mapped from the text:
    - Find number of subject associations;
    - Find number of all associations;
    - Calculate subjectivity ratio:

$$subj\_ratio = \frac{\#\ of\ subject\ associations}{\#\ of\ all\ associations}$$

    - Calculate subjectivity rank:

$$subj\_rank = concept\_weight * subj\_ratio$$

  - Pick top-n subjects with highest subjectivity rank.

Figure 5: Process of candidate subjects discovery.

Next, the system (ii) creates new English sentences for the candidate subjects. To generate new sentences system uses subject–predicate–object structure enhanced with the adjective modifiers for subjects and objects, and the adverb modifiers for predicates, when available. Subject, predicate and object elements are mandatory while adjective and adverb modifiers are optional. The system chooses candidate elements for the sentence using the weight of the association between the concepts. Created sentences form final summary of a given text. Sub-process is outlined in Figure 6.

(ii) New sentence generation sub-process:

○ For each subject in top-n subjects:
- Convert subject Cyc concept to natural language representation **(a)**;
- Pick adjective with highest subject-adjective association weight;
- Convert adjective Cyc concept to natural language representation **(b)**;
- Pick top-n predicates with highest subject-predicate association weights;
- For each predicate in top-n predicates:
  - ▪ Convert predicate Cyc concept to natural language representation **(c)**;
  - ▪ Pick adverb with highest predicate-adverb association weight;
  - ▪ Convert adverb Cyc concept to natural language representation **(d)**;
  - ▪ Pick top-n objects with highest product of subject-object and predicate-object associations weights;
  - ▪ For each object in top-n objects:
    - Convert object Cyc concept to natural language representation **(e)**;
    - Pick adjective with highest object-adjective association;
    - Convert adjective Cyc concept to natural language representation **(f)**;
    - Create new sentence using subject **(a)**, subject-adjective **(b)**, predicate **(c)**, predicate-adverb **(d)**, object **(e)**, object-adjective **(f)** natural language phrases.

Figure 6: Process of new sentence generation.

7 IMPLEMENTATION AND TESTING

We implemented the system in Python programming language. Python was a natural choice because of the advanced Natural Language Processing tools and libraries supplied by the language. Cyc knowledge base supports inference engine operations through SubL language commands and Java APIs. Some of the SubL commands we used were “min-genls” to find concept’s ancestors, “generate-phrase” to convert Cyc concepts to natural language representation and “query-variable” to run queries against the knowledge base. We used Java-Python wrapper implemented by JPype library (JPype, 2017) to communicate with Cyc server and perform reasoning. The system was design pipelined and modular to allow comprehensible data flow and convenient maintenance.

We conducted several experiments to highlight different capabilities of the system. First, we applied the system on Wikipedia articles describing concepts from different domains. The articles described domestic dog, hamburger and personal computer. Table 1 shows main topics and concepts extracted from the analysed articles. Topics are represented by the micro theories from Cyc knowledge base. For example, #$BiologyMt micro theory contains general information about the living things; #$HumanFoodGMt micro theory describes human food; #$HumanSocialLifeMt micro theory covers social and cultural aspects of human relationships. Concepts are represented as the Cyc terms. Each term has a natural language representation, e.g. “canis” for #$CanisGenus, “subspecies” for #$BiologicalSubspecies and “developer” for #$ComputerProgrammer.

Some of the new sentences created for the articles are outlined in Figure 7. All sentences have minimal subject-predicate-object structure and some of the sentences go beyond with additional adjective and adverb modifiers. This is possible when subject, predicate or object has strong adjective or adverb relations.

“*Dog being canis.*”
“*Dog having short external anatomic part.”*

“*Burger utilizing traditional mammal meat.*”
“*Ground beef being bovine meet.*”

“*Computer having computer program.*”
“*Computer hardware needing power.*”

Figure 7: Test results of some of the new sentences created for Wikipedia articles.

Next, we conducted experiment using multiple articles about grapefruit. New sentences created by the system are outlined in Figure 8. These results show the progression from subject-predicate-object

Table 1: Test results of main topics and concepts derived from Wikipedia articles.

Article: *Dog*	**Article**: *Hamburger*	**Article:** *Computer*
Topics (micro theories): • *#$BiologyMt* • *#$BiologyVocabularyMt* • *#$NaivePhysicsVocabularyMt*	**Topics (micro theories):** • *#$HumanFoodGMt* • *#$HumanFoodGVocabularyMt* • *#$ProductGVocabularyMt*	**Topics (micro theories):** • *#$InformationTerminologyMt* • *#$HumanSocialLifeMt* • *#$NaivePhysicsVocabularyMt*
Concepts: • *#$Dog* • *#$CanisGenus* • *#$Person* • *#$BiologicalSubspecies* • *#$Breeder*	**Concepts:** • *#$Food* • *#$Burger* • *#$HamburgerSandwich* • *#$GroundBeef* • *#$Cheese*	**Concepts:** • *#$Computer* • *#$ComputerProgrammer* • *#$outputs* • *#$ComputerHardwareItem* • *#$ControlDevice*

structure to more complex structure extended by the adjective and adverb modifiers when more articles were processed by the system.

"*Grapefruit being fruit.*" (a)

"*Grapefruit being colored edible fruit.*" (b)

"*Colored grapefruit being sweet edible fruit.*" (c)

Figure 8: Test results of new sentences created for multiple articles about grapefruit; (a) – single article, (b) – two articles, (c) – three articles.

Finally, we have applied the system on five Wikipedia articles describing different types of felines (cat, tiger, cougar, jaguar and lion). Table 2 shows main topics and concepts extracted from the text and new created sentences.

Test results show that the system is able to create sentences that contain generalized concepts and combine information from different parts of the text. Concepts like "canis", "mammal meat" and "felis" were derived by the abstraction process and were not mentioned in the original text. The system yields better results compared to the reported in (Choi & Huang, 2010). New sentences created by our system have structure that is more complex and contain information fused from various parts of the text.

8 CONCLUSION AND FUTURE WORK

In this paper, we described a knowledge based automatic summarization system that creates an abstractive summary of the text. This task is still challenging for machines, because in order to create such summary, the information from the input text has to be aggregated and synthesized, drawing knowledge that is more general. This is not feasible without using the semantics and having domain knowledge. To have such capabilities, our described system uses Cyc knowledge base and its reasoning engine. Utilizing semantic features and syntactic structure of the text shows great potential in creating abstractive summaries.

We have implemented and tested our proposed system. The results show that the system is able to abstract new concepts not mentioned in the text, identify main topics and create new sentences using information from different parts of the text.

We outline several directions for the future improvements of the system. The first direction is to improve the domain knowledge representation, since the semantic knowledge and reasoning are only limited by Cyc knowledge base. Ideally, the system

Table 2: Test results of new sentences, concepts and main topics for Wikipedia articles about felines.

Topics (micro theories):	**Concepts:**	**Sentences:**
• *#$BiologyMt* • *#$BiologyVocabularyMt* • *#$HumanSocialLifeMt*	• *#$Cat* • *#$DomesticCat* • *#$FelisGenus* • *#$FelidaeFamily* • *#$Animal*	"*Cat usually being native animal.*" "*Big felis usually being natural predatory animal.*" "*Big felis usually being exotic animal.*" "*Big felis often using killing method.*" "*Big felis often using marking.*" "*Male feline often killing prey.*" "*Male feline living historical mountain range.*"

would be able to use the whole World Wide Web as a domain knowledge, but this possesses challenges like information inconsistency and sense disambiguation. The second direction is to improve the structure of the created sentences. We use subject-predicate-object triplets extended by adjective and adverb modifiers. Such structure can be improved by using more advanced syntactic representation of the sentence, e.g. graph representation. Finally, some of the created sentences are not conceptually connected to each other. Analysing the relations between concepts on the document level will help in creating sentences that will be linked to each other conceptually.

REFERENCES

Barzilay, R. & Elhadad, M., 1999. Using lexical chains for text summarization. *Advances in automatic text summarization*, pp. 111-121.

Bellare, K. et al., 2004. *Generic Text Summarization Using WordNet.* Lisbon, Portugal, LREC, pp. 691-694.

Bing, L. et al., 2015. *Abstractive multi-document summarization via phrase selection and merging.* Beijing, China, Association for Computational Linguistics, pp. 1587-1597.

Cheung, J. C. K. & Penn, G., 2013. *Towards Robust Abstractive Multi-Document Summarization: A Caseframe Analysis of Centrality and Domain.*. Sofia, Bulgaria, Association for Computational Linguistics, pp. 1233-1242.

Choi, B. & Huang, X., 2010. *Creating New Sentences to Summarize Documents.* Innsbruck, Austria, IASTED, pp. 458-463.

Cycorp, 2017. *Cycorp – Cycorp Making Solutions Better.* [Online]
Available at: http://www.cyc.com/
[Accessed July 2017].

Ganesan, K., Zhai, C. & Han, J., 2010. *Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions.* Beijing, China, Association for Computational Linguistics, pp. 340-348.

Gong, Y. & Liu, X., 2001. *Generic text summarization using relevance measure and latent semantic analysis.* New Orleans, Louisiana, ACM, pp. 19-25.

Günes, E. & Radev, D. R., 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, Issue 22, pp. 457-479.

Honnibal, M. & Johnson, M., 2015. *An Improved Non-monotonic Transition System for Dependency Parsing.* Lisbon, Portugal, Association for Computational Linguistics, pp. 1373-1378.

Hovy, E. & Lin, C.-Y., 1998. *Automated text summarization and the SUMMARIST system.* Baltimore, Maryland, Association for Computational Linguistics, pp. 197-214.

JPype, 2017. *JPype - Java to Python integration.* [Online]
Available at: http://jpype.sourceforge.net/
[Accessed July 2017].

Luhn, H. P., 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2), pp. 159-165.

Matuszek, C., Cabral, J., Witbrock, M. & DeOliveira, J., 2006. *An Introduction to the Syntax and Content of Cyc.* Palo Alto, California, AAAI, pp. 44-49.

Mihalcea, R. & Tarau, P., 2004. *TextRank: Bringing Order into Text.* Barcelona, Spain, EMNLP, pp. 404-411.

Moawad, I. F. & Aref, M., 2012. *Semantic graph reduction approach for abstractive Text Summarization.* Cairo, Egypt, IEEE, pp. 132-138.

Nallapati, R., Zhai, F. & Zhou, B., 2017. *SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents.* San Francisco, California, AAAI.

Nenkova, A. & McKeown, K., 2012. A survey of text summarization techniques. In: C. C. Aggarwal & C. Zhai, eds. *Mining Text data.* s.l.:Springer, pp. 43-76.

Pal, A. R. & Saha, D., 2014. *An approach to automatic text summarization using WordNet.* Gurgaon, India, IEEE, pp. 1169-1173.

Radev, D. R., Jing, H., Styś, M. & Tam, D., 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6), pp. 919-938.

Rodriguez, A. & Laio, A., 2014. Clustering by fast search and find of density peaks. *Science*, 344(6191), pp. 1492-1496.

Rush, A. M., Chopra, S. & Wetson, J., 2015. *A neural attention model for abstractive sentence summarization.* Lisbon, Portugal, EMNLP.

Shen, D. et al., 2007. *Document Summarization Using Conditional Random Fields.* Hyderabad, India, IJCAI, pp. 2862-2867.

Wong, K.-F., Wu, M. & Li, W., 2008. *Extractive summarization using supervised and semi-supervised learning.* Manchester, United Kingdom, Association for Computational Linguistics, pp. 985-992.

Ye, S., Chua, T.-S., Kan, M.-Y. & Qiu, L., 2007. Document concept lattice for text understanding and summarization. *Information Processing & Management*, 43(6), pp. 1643-1662.