EVALUATION TECHNIQUES AND GRAPH-BASED ALGORITHMS FOR AUTOMATIC SUMMARIZATION AND KEYPHRASE EXTRACTION

Fahmida Hamid

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

August 2016

APPROVED:

Paul Tarau, Major Professor
Rada Mihlacea, Committee Member
Bill Buckles, Committee Member
Eduardo Blanco, Committee Member
Bryant R. Barrett, Chair of the Department of
Computer Science and Engineering
Costas Tsatsoulis, Dean of the
College of Engineering
Victor Prybutok, Vice Provost of the
Toulouse Graduate School
University of North Texas

Hamid, Fahmida. *Evaluation Techniques and Graph-based Algorithms for Automatic Summarization and Keyphrase Extraction*. Doctor of Philosophy (Computer Science and Engineering), August 2016, 86 pp., 23 tables, 9 figures, 74 numbered references.

Automatic text summarization and keyphrase extraction are two interesting areas of research which extend along natural language processing and information retrieval. They have recently become very popular because of their wide applicability. Devising generic techniques for these tasks is challenging due to several issues. Yet we have a good number of intelligent systems performing the tasks. As different systems are designed with different perspectives, evaluating their performances with a generic strategy is crucial. It has also become immensely important to evaluate the performances with minimal human effort.

In our work, we focus on designing a relativized scale for evaluating different algorithms. This is our major contribution which challenges the traditional approach of working with an absolute scale. We consider the impact of some of the environment variables (length of the document, references, and system generated outputs) on the performance. Instead of defining some rigid lengths, we show how to adjust to their variations. We prove a mathematically sound baseline that should work for all kinds of documents. We emphasize automatically determining the syntactic well-formedness of the structures (sentences). We also propose defining an equivalence class for each unit (e.g. word) instead of the exact string matching strategy. We show an evaluation approach that considers the weighted relatedness of multiple references to adjust to the degree of disagreements between the gold standards. We publish the proposed approach as a free tool so that other systems can use it. We have also accumulated a dataset (scientific articles) with a reference summary and keyphrases for each document. Our approach

is applicable not only for evaluating single-document based tasks but also for evaluating multiple-document based tasks.

We have tested our evaluation method for three intrinsic tasks (taken from DUC 2004 conference), and in all three cases, it correlates positively with ROUGE. Based on our experiments for DUC 2004 Question-Answering task, it correlates with the human decision (extrinsic task) with 36.008% of accuracy. In general, we can state that the proposed relativized scale performs as well as the popular technique (ROUGE) with flexibility for the length of the output.

As part of the evaluation we have also devised a new graph-based algorithm focusing on sentiment analysis. The proposed model can extract units (e.g. words or sentences) from the original text belonging either to the positive sentiment-pole or to the negative sentiment-pole. It embeds both (positive and negative) types of sentiment-flow into a single text-graph. The text-graph is composed with words or phrases as nodes, and their relations as edges. By recursively calling two mutually exclusive relations the model builds the final rank of the nodes. Based on the final rank, it splits two segments from the article: one with highly positive sentiment and the other with highly negative sentiments. The output of this model was tested with the non-polar TextRank generated output to quantify how much of the polar summaries actually covers the fact along with sentiment.

ACKNOWLEDGMENTS

I owe a great deal of gratitude to many who have supported me in all possible ways during this part of my life.

I would like to thank the almighty Allah for giving me this opportunity to study in USA that most of the students from a third world country cannot afford. Then my special thanks and gratitude goes to my parents. My mother was the first person who whole heartedly supported me with my decision of starting the Ph.D.

I would also like to thank my advisor, Paul Tarau, for his clear guidance, and invaluable helps. He was always available to help me with advice and encouragement. He always became the first critic of any article I wrote or any research idea I came up with, and thus helped me enhancing my skill and knowledge. He showed great deal of patience and did let me play with a few areas before picking the one I preferred the most. I would also like to thank all my committee members for their support, and care.

I really need to thank a few special persons without who living in Denton would have become exhaustive. Khadiza Zaman, Mahzabeen Islam, and Syed Arif Ali are those precious persons to whom I will not be able to express enough gratitudes. I would also like to thank my brother, Kaiser Hamid, for being on my side and always giving me the positive feedback at the time of need.

Last but not the least, I express my heartiest thanks to the SELL lab members with who I have been sharing the lab, some thoughts, and some fun-time for the last four years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

BEGINNING THE JOURNEY

1.1. Introduction

Information retrieval from text data is a demanding area of study. Automatic summarization and keyphrase extraction are two information retrieval tasks which incorporate many important aspects of both natural language understanding and natural language generation [60]. These area are interesting as well as challenging. Humans need to understand the language and topic(s) described in the article(s) to perform these tasks properly. Sometimes humans use external knowledge sources to improve their results. It is quite common that a native speaker of any particular language is more comfortable with these tasks than a non-native speaker. Therefore, it is easily understood how difficult the task would be for a machine or system. On the contrary, with the exponential growth-rate of data, performing these tasks with human-laborers are extremely expensive and time consuming; in short, it is infeasible. In order to compensate this scenario, machines (intelligent systems, computer programs) are the alternatives we have.

The goal of text summarization is to take a textual document, 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 [38]. The web search engines, for example, have exploited the use of text summarization from the very beginning: starting with the extraction of a certain number of bytes to the more sophisticated query focused summaries typified by Google's snippet [73]. Keyphrases, on the other hand, provide semantic meta-data to characterize the document. Keyphrases are useful because they briefly summarize a document's content. As large document collections such as digital libraries become widespread, the value of such summary information increases. Keywords and keyphrases are particularly useful because they can be interpreted individually and independently of each other. They can be used in information retrieval systems as descriptions of the documents returned by a query, as the basis for search indexes, as a way of browsing a collection, and as a document clustering technique [70].

Automatic summarization is a more diverse problem than keyphrase extraction or generation. We can categorize the summaries in various ways based on different situations. Summaries differ according to what they extract or abstract. Summaries can either be domain-specific or generic. A domain-specific summary is assumed to contain idiosyncratic words. No such assumption can be made for the generic summary. A fluent summary is written using well-formed grammatical sentences which are related and follow one another according to the rules of coherent discourse structure. A non-fluent summary, on the other hand, is composed with fragmented units, such as phrases. A query-oriented summary favors specific themes or aspects. An informative summary reflects the content, whereas an indicative one merely includes the content excepting the dominant topic. Every kind of summary has two core tasks: determining what is salient in the source and deciding how to reduce the content. But within and across the categories, summaries differ according to function and target reader. For example, a summary can be indicative, informative, or critical based on the purpose of the summary. A better understanding of the types of the summary will facilitate the design of new algorithms with improved performances. Our discussion in this paragraph is heavily motivated by automatic text summarization articles such as [16, 13].

Keyphrases and summaries can either be extractive or abstractive. The extractive approaches select the salient pieces from the original source and concatenate them to yield a shorter version. An abstractive approach, however, paraphrases using more generic terms. A person with moderate control over the language is expected to generate a set of qualitative keyphrases (or, write an informative and meaningful summary). Since natural language changes over time and has innumerable ways to express a statement, it is very difficult for a system to generate a synopsis of the article that is as readable as the human written synopsis. Language generation (at least paraphrasing, sentence-fusion, and word-replacement) without introducing ambiguity is one of the most challenging problems so far. This is the main reason that extractive approaches are more popular than abstractive approaches. Summaries and keyphrases can be generated from a single document or multiple documents. For example, to cluster a set of books under a topic or to write a follow-up story on an incident, one uses multiple documents. Working with multiple documents has some concerns (e.g. removing redundancy, maintaining the flow of information, coving all or dominant topic(s), etc.) which are less intense with single documents.

Both supervised and unsupervised approaches are popular among researchers while designing an algorithm for summarization and keyphrase-extraction. Supervised approaches achieve comparably better performance if they are trained with appropriate features. They need a moderately large training dataset with the human generated reference-set(s) for feature extraction and parameter estimation. The approaches may suffer if the features are domain-dependent, and systems are trained in one domain and tested over another. They also suffer due to the potential inconsistency of human generated outputs. It is strenuous and expensive to generate a standard training set with sufficient references. Unsupervised approaches are less susceptible to the domains but their evaluation-score still depends on the quality of human provided references.

Many text summarization methods are surveyed in [38, 36] and in the Document Understanding Conference (which is known as the Text Analysis Conference since 2008). From the established and well-known approaches, we can outline some important techniques that are helpful in almost all cases. Maybury et. al [36] categorized text summarization approaches into surface, entity, and discourse levels. Surface level approaches represent information with shallow features, e.g. term-frequency (tf), inverse document frequency (idf), sentence position, cue word, etc., and they combine these features to yield a salience function that measures the significance of information [71]. Entity level approaches use different entities (word, sentence, paragraph, etc.) and their relationships (e.g. co-occurrence, co-reference) to model the text and extract the most important segments. Discourse level approaches use rhetorical structure of the document and its relation to the communicative goals. For example, summarizing a scientific article will be highly beneficial if sentences are extracted from specific sections like the conclusion, introduction, or result-analysis. However, most IR (information retrieval) techniques that have been exploited in text summarization focus on symbolic level analysis, and do not take into account semantics such as synonymy, polysemy, and term dependency [16].

A great deal of work is ongoing to leverage the performance of the systems. Hovy and Lin [16], for example, attempted to create a summarization system based on the equation:

$$summarization = topic\text{-}identification + interpretation + generation$$

Mihalcea and Tarau [45] have proposed a graph-based ranking model for text processing, and they have shown how this model can be successfully used in natural language applications (especially in summarization and keyphrase extraction). Erkan et al. [7] introduced a stochastic graph-based method for computing relative importance of the textual units for natural language processing. Radev et al. [59] proposed a multi-document summarizer (MEAD) which generates summaries using cluster-centroids produced by a topic detection and tracking (TDT) system. Authors also used some phrase matching techniques to detect cross sentence information subsumption to control the redundant information. Carbonell [4] used the Maximal Marginal Relevance criteria to reduce the redundancy while maintaining query relevance in re-ranking documents and in selecting appropriate passages for text summarization. We need to carefully consider the context factor and the purpose factor to design a summarization system. When the range of summarization contexts is considered, there is no reason to suppose that any one summary, even a supposedly good one, would meet all the context constraints [22]. So, it is beneficial to have multiple reference summaries for each document to compare against the system generated outputs. However to afford them for a collection of millions of articles is a challenging issue.

When we have multiple references the degree of agreement or disagreement between the references becomes one of the vital affairs. The absolute scales (precision, recall, f-score) or their variants usually combine all of the available references together and find the overlap with the system generated output. We believe, the agreement between the evaluators should be handled in a more delicate manner. We explain the validity of the scenario with two examples taken from two different evaluation conferences(SemEval, and Document Understanding Conference). For example, author assigned keyphrases for a research article differ to some extent from the reader assigned keyphrases (Table 1.1). From the table, we also find that the number of outputs generated by the readers are different from the number of outputs generated by the author of the article. Another interesting fact is, the evaluators (author and reader) agree exactly on one of the keyphrases out of several chosen ones though a few of them are semantically related. In case of summarization, a human can generate slightly different summaries for the same document, if asked to do the job at different times or in a different order. Also, humans can improvise new phrases indicating the same event. For example, in Table 1.2, three phrases, {out of country, Beijing, abroad}, are used by the evaluators (A, B, C, and D) to indicate the same idea.

Table 1.1. Gold-keyphrases for document C-44 of SemEval-2010, task-5

author-assigned	reader-assigned
wireless sensor network	multi-sequence positioning
localization	wireless sensor network
node sequence process	massive uva-based deploment
	node localization
	spatiotemporal correlation
	event distribution
	range-based approach
	distribution-based location estimation
	listen-detect-assemble-report protocol
	marginal distribution

Table 1.2. Reference summaries for document APW19981016.0240 @ DUC-2004, task-1

reference id	summary
A	Cambodian government rejects opposition's call for talks abroad
B	Cambodian leader Hun Sen rejects opposition demands for talks in Beijing.
C	Hun Sen rejects out of country talks, Sihanouk asked to host summit
D	New Cambodian government in limbo as Hun Sen rejects talks out of country

After carefully considering the discussed cases, we can state that designing a common framework for evaluating the performances of different systems is a very challenging task. We outline some major issues from the discussions so far:

- We should consider different contextual aspects (actual words in the documents and in the references, the semantic relatedness between different words, the context of the words, the length of the original document, etc.) while developing an evaluation strategy.

- A length-independent scale could be a better alternative to the traditional absolute approach.
- At the presence of multiple gold standards, the references should be weighted by comparing the relatedness among themselves.
- The evaluation approach should be easily modifiable to adjust to the assigned tasks. For example, evaluating domain-specific summarization might consider a few other matters than evaluating a generic summarization task. In an ideal case, a robust evaluation strategy is supposed to take care of all the situations by adjusting relevant parameters.

In the following section we state our research questions and goals that we have set for the Ph.D. journey.

1.2. Research Topics

We started to work with different graph-based techniques for generating summaries and keyphrases. Eventually we ran into some challenges while trying to evaluate our designed approach with existing state-of-the-art techniques. Therefore we will be addressing two related tracks as our research topic. Our prime idea is to focus on designing a generic evaluation technique for summary and keyphrases. The subsidiary track is to devise some graph-based algorithms for summary extraction. We consider the second research track as a case study of extrinsic summarization task and we apply our evaluation technique to verify the informativeness of the graph-based algorithm generated output.

1.2.1. Topic I: A Generic Evaluation Technique for Summary and Keyphrases

In order to evaluate the performance of some algorithms we need a moderately large dataset and reliable standard outputs (known as gold standards). Manually labeling the datasets and generating base-cases and gold standards for comparison is infeasible, and time consuming. We need effective techniques to minimize the manual outputs needed for the evaluation process. A recent study [10] states that at least 17.5 hours of human labor are needed to evaluate two systems that follow the usual TAC (Text Analysis Conference) structure (around 500 articles, and four reference summaries per article). It is also hard to standardize the results with a single human annotator. There is always some degree of disagreements between humans for most of the tasks.

Summaries are tailored to a reader's interest and expertise, yielding topic-related summaries, or else they can be aimed at a broad readership community, as in the case of generic summaries [37]. An intrinsic or normative evaluation judges the quality of the summary directly based on analysis in terms of a set of norms. An extrinsic evaluation, on the other hand, judges the quality of the summarization based on how it affects the completion of some other task. An important parameter to summarization is the level of compression. In the TIPSTER SUMMAC (1999) conference, it was reported that summaries as short as 17% of full text-length sped up decision making by almost a factor of 2. From the set of outputs of the Document Understanding Conference (DUC), 2004, it is noticeable that a summary created with very high compression ratio is hardly distinguishable from a set of keyphrases. It usually does not contain syntactically correct sentences; rather some phrases connected through punctuation.

We believe a system generated summary should also follow syntactic rules to form the basic units (sentences). Thus, it is important to consider not only the content overlapping between the references and the system output, but also to evaluate the "syntactic well-formedness". Besides, comparing two summaries is sensitive to their lengths and the length of the document they are extracted from. It is worth considering this issue and developing an automatic evaluation technique that can adjust to the length variation.

While humans, by nature, choose abstractive outputs over extractive ones, most of the systems follow extractive algorithms. So, there will always be some asymmetries between the expected set and the extracted set. With this idea in mind, we develop an evaluation technique that forms a relativized scale and extend it to consider semantic information in the evaluation process.

To weigh gold standards with some discrepancy is a challenge to the evaluation approaches. Evaluation based on fixed lengths and averaging the number of overlapped units with respect to total units is an old technique (usually named "absolute scale"). It is time to redefine the absolute scales with a more flexible relativized approach, and design a generic "baseline" which is proven to be mathematically sound. We also need to "evaluate the evaluators" and thus design a scaling mechanism for evaluating summaries and keyphrases.

1.2.2. Topic II: A Case Study on Sentiment-oriented Summary Generation and Evaluation

Automatic summarization and keyphrase extraction are two well-known approaches to distill the most important information from a set of sources. These days with the expanding nature of the World Wide Web, summarization and keyphrase extraction from documents based on related topics, questions, or queries have become more demanding. Besides focusing on local text-based statistical information, researchers are extracting related information from available knowledge-bases to fine-tune their algorithms. Cross-lingual models are also being used to disambiguate some terms and to relate one with the other. Despite the researchers' countless efforts, some results are far from human-like quality. We find it interesting and necessary to expand existing algorithms with the aim of improvement in performance, and at the same time, to re-design the existing evaluation techniques to consider certain facts. We have worked on the several projects such as

- re-using a graph-based algorithm (initially designed to symbolize a trust-based peer-network) to extract sentiment-polar summaries from each document (discussed in Chapter 5).
- designing a framework based on the trust-based model to automatically generate context-aware sentiment-lexicon per topic (discussed in Chapter 6).
- expanding the topic-based TextRank algorithm with some context information to extract topic-focused keyphrases from scientific articles.

Regardless of using different sources of external information, the outcomes are marginal. Hence we start using some simple techniques to regenerate, i.e. reform the results that we have from extractive approaches.

1.3. Research Goals

We will try to place our goals from both of the research tracks under the same umbrella so that the connection between them becomes obvious to the reader.

- To design a more appropriate and efficient evaluation technique for summary and keyphrase extraction tasks is our principal area of research. We aim to resolve some flaws or issues that are very common with the existing absolute scales. The synopsis of our research goals are the following ones:

  - to define a mathematically sound baseline
  - to suggest a relativized scale that takes into account different outputs with possibly different lengths
  - to design a methodology that carefully considers the degree of agreement between the human evaluators
  - to embed a semantic knowledge-base with the evaluation process so that the extractive and the abstractive approaches become comparable with each other
  - to accredit syntactic well-formedness as well as informativeness
  - to create a large database and at least one reliable reference set that can be useful for both of the tasks

- We then apply the evaluation technique for an extrinsic summarization task [15] to verify the proposed process. This way, we show an application of the evaluation approach on a dataset that has no prior human-evaluations from the expected point of view.

1.4. Structure of the Thesis

In this thesis, we present our research, achievements, evaluations, and future directions through the following chapters:

Chapter 1: The initial Chapter states the importance and relevance of our research direction.

Chapter 2: In this Chapter, we introduce our major research idea that can be applied to evaluate summaries and keyphrases.

Chapter 3: This Chapter states how to adopt our evaluation scale with the presence of multiple references.

Chapter 4: We introduce a dataset and provide an extension of our evaluation approach to accommodate abstractive summaries and keyphrases.

Chapter 5: In this Chapter, we show a case study where we introduce a refined version of a graph-based model to extract sentiment-polar summaries. We also use our evaluation methodology to test the informativeness of the extracted summaries.

Chapter 6: In this Chapter, we detail a framework to generate topic-specific sentiment lexicons using the graph-based technique that has shown effective performance in the case study (Chapter5).

Chapter 7: Finally, we state the concluding remarks and future directions of our work.

CHAPTER 2

A RELATIVIZED SCALE FOR SUMMARY AND KEYPHRASE EVALUATION

If you are to trust the summary is indeed a reliable substitute for the source, you must be confident that it does in fact reflect what is relevant in the source. Hence, methods for creating and evaluating summaries should complement each other [13].

2.1. Introduction

The evaluation on an NLP system is a key part of any research or development effort and yet it is probably the most controversial [20]. Accurate computer-based evaluation of system-generated summaries (or keyphrases) is far from being obvious or easy. Most of the shortcomings might come from the simplifications that statistical measures need to assume. The existing evaluation approaches use absolute scales (e.g. precision, recall, f-measure) to evaluate the performance of the participating systems. These measures can be used to compare the informativeness of the produced outputs, but they do not indicate how significant one is from the other. The IR community's hard-won experience shows there are no easy ways of evaluating systems, no magic numbers encapsulating performance, no 'core' functions that can be pursued far in isolation, no fixed meaning-representation devices any system must have [21]. As evaluating summaries can be seen as a more complicated problem than evaluating keyphrases, we will be discussing the evaluation-related issues mostly for summaries. And, at some later section, we will map the same methodology to evaluating keyphrase extraction techniques.

A summary evaluation methodology can be characterized considering two major categories: intrinsic and extrinsic. An evaluation of system generated summary against an ideal summary is the intrinsic approach. The evaluation of how well summaries help a person perform in a task is defined in extrinsic approach. In the evaluation process, it is essential to take both system and environment into account as they supply the factors affecting performance. Most of the time we pay too much attention to the system, but not enough to the environment. The environment, for example, is composed with the given documents, the language, the structure of the document, type of task (intrinsic or extrinsic), length of the document, compression and retention ratio, and so on.

Evaluating the quality of a summary has been proven to be a difficult task as there is no obvious “ideal” summary. Even for relatively straightforward news articles, human summarizers tend to agree only approximately 60% of the time, measuring sentence content overlap [57]. In content based evaluation, system output is compared sentence by sentence or fragment by fragment to one or more human-made ideal abstracts, and as in information retrieval the percentage of extraneous information present in the system's summary (precision) and the percentage of the important information omitted from the summary (recall) are recorded.

$$precision = \frac{|\{\text{relevant document}\} \cap \{\text{retrieved document}\}|}{|\{\text{retrieved document}\}|}$$

$$recall = \frac{|\{\text{relevant document}\} \cap \{\text{retrieved document}\}|}{|\{\text{relevant document}\}|}$$

In order to quantify the performance with a single score, we frequently use balanced harmonic mean (f-$measure$) of precision and recall.

$$f\text{-}measure = \frac{2 * precision * recall}{precision + recall}$$

Other common measures include $Kappa$ [5] and relative utility [59]. The kappa coefficient (K) measures pairwise agreement among a set of coders making category judgments, correcting for expected chance agreement:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ is the proportion of times that the coders agree and $P(E)$ is the proportion of times that we would expect them to agree by chance, calculated along the lines of the intuitive argument presented above. In the relative utility based measure ($CBSU$), the normalized system performance D is expressed as a linear function relating human-response J,systen-response S and random performance R.

$$D = \frac{S - R}{J - R}$$

But it is worth noting that the random performance R is the average of all system outputs at a given compression rate. So, it is related to the participating system's output, and the provided reference samples. It has no direct connection to the environment (at least to the original document).

Efforts have also been given to designing an evaluation metric without using human references, such as [33]. The underlying intuition was that a good summary will tend to be similar to the input document in terms of content. Though the hypothesis sounds quite interesting, it is not very clear how similarity-measures should be defined for this particular problem. Measures of similarity (e.g. Kullback Leibler divergence, Jensen Shannon divergence, cosine similarity) between two probability distributions is a natural choice. The Jensen Shannon divergence incorporates the idea that the distance between two distributions (P and Q, assuming one as the document, and the other as the system generated summary) cannot be very different from the average of distances from their mean distribution. It can be defined with the following equation:

$$J(P||Q) = \frac{1}{2}[D(P||A) + D(Q||A)]$$

where $A = \frac{P+Q}{2}$ is the mean distribution of P and Q. The JS-divergence is symmetric and always defined. It is good that it considers at least one feature of the environment in the evaluation process. On the other hand, if we look at the standard summary evaluation conferences (DUC, TAC, etc.) and the participants' output samples, we find that most of the system outputs are really a few bytes of words, not even a subset of complete sentences. Also, most of them followed extractive approach. Hence, the impact of using probability distribution does not introduce much variation on the evaluation process. We would like to refer to the output shown at Table 1 & 2 of [29] as an indicator of our statement.

Another common issue while evaluating summaries (or any task) is defining the baseline (i.e. the lower bound) of any task. Various approaches are been tried. For example, the TAC 2011 included two baseline summarizers:

- Summarizer 1: returns all the leading sentences (up to 100 words) in the most recent document. Summarizer 1 provides a lower bound on what can be achieved with a simple fully automatic extractive summarizer.
- Summarizer 2: output of MEAD [7] automatic summarizer (v 3.12, publically available at `http://www.summarization.com/mead/`), with all default settings, set to producing 80-word summaries (which, due to MEAD's non-strict word limit, in effect pro-

duced summaries consisting of complete sentences under the required 100 words limit).

While evaluating with an absolute scale, and even with some adjustments by Kappa, CBSU, or probability distribution, we control/trim the length of the output according to the reference's length to get a fair scenario. Then, we either use an exact phrase/word matching technique or have to employ manual labor for generating paraphrases. Considering the existing approaches, we can state that the absolute scales suffer due to the length constraints. Comparing two summaries is sensitive to their lengths and the length of the document they are extracted from. The statement holds for keyphrase extraction task as well. We, therefore, propose a relativized scale (i-$measure$) [14] with some weighted-matching strategy that is suitable for evaluating both of the tasks. We, then, propose a modified approach of i-$measure$ that not only can adjust to the length variation but also considers an equivalence-relation using WordNet [48] provided synsets to modify the "observed intersection (matching) size".

2.2. Popular Evaluation Techniques

2.2.1. Summary Evaluation

The process of summarization can be decomposed into three major phases: analysis, transformation, and synthesis. The analysis phase analyzes the input text and selects a few salient features. The transformation phase transforms the results of analysis into a summary representation. The synthesis phase prepares an appropriate summary according to the specified task using the output from transformation phase. In the overall process, the compression rate, which is defined as the ratio between the length of the summary and that of an original, is an important factor [71].

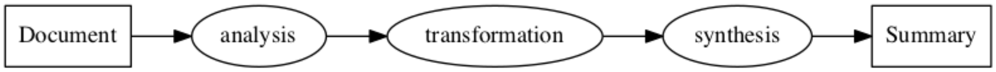


Figure 2.1. Steps of "Summarization"

ROUGE [28] is one of the well known techniques to evaluate single or multi-document summaries. ROUGE includes measures to automatically determine the quality (or, content) of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the machine-generated summary and the reference summaries.

$$rouge\_score = \frac{\sum_{S \in ReferenceSummaries} \sum_{gram_n \in S} count_{match}(gram_n)}{\sum_{S \in ReferenceSummaries} \sum_{gram_n \in S} count(gram_n)}$$

$$pyramid\_score = \frac{sum\ of\ weights\ of\ SCUs\ expressed\ in\ S}{sum\ of\ weights\ of\ ideal\ summary\ with\ the\ same\ number\ of\ SCU\ as\ S}$$

Another summary evaluation tool Pyramid [51] considers multiple models to build a gold standard for system output. Each tier of the pyramid quantitively represents the agreements among human summaries based on Summary Content Units (SCU). SCUs are not bigger than a clause. SCUs that appear in more of the human summaries are weighted more highly, allowing differentiation between important content from less important one. The original pyramid score is similar to a precision metric. And the pyramid layers are defined with the concept of multiple references. Hence, it has no impact if there is only a single reference. Initially the SCUs were defined by humans, which is a restriction to designing a completely automatic evaluation tool.

2.2.2. Keyphrase Evaluation

The evaluation of keyphrase extraction, on the other hand, has not received much research attention so far. In most of the cases, researchers use the precision at k (where, $k \in \{5, 10, 15\}$) to report their systems' performance. If multiple references are available, there is no clear direction to treat them. For example, in ROUGE, all the references for a particular document (or, document-cluster in case of multi-document based tasks) are combined as a single set. Another approach can be to weigh the references against one-another based of some criteria. The other important issue is, the reference set does not always contain exactly same number of keyphrases as the system generated sets. Hence, R-p [72] (a deviation from pure precision) was proposed by authors to chop various lengths of outputs to a single value.

The idea of R-p was taken from [64]. In information retrieval, R-p is the precision when the number of retrieved documents equals the number of relevant documents. It requires having a set of known relevant documents Rel, from which we calculate the $precision$ of the top Rel documents returned. (The set Rel may be incomplete, such as when Rel is formed by creating relevance judgments for the pooled top k results of particular systems (S) in a set of experiments.)

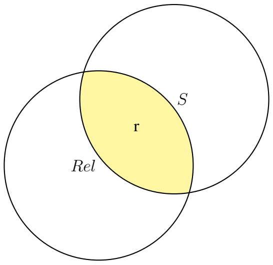


Figure 2.2. A set, Rel of related outputs where $|Rel| = |S|$

R-$precision$ adjusts for the size of the set of relevant documents: A perfect system could score 1 on this metric for each query, whereas, even a perfect system could only achieve a precision at 20 of 0.4 if there were only 8 documents in the collection relevant to an information need. Averaging this measure across queries thus makes more sense. If there are $|Rel|$ relevant documents for a query, we examine the top $|Rel|$ results of a system, and find that r are relevant, then by definition, not only is the precision (and hence R-precision) $\frac{r}{|Rel|}$, but the recall of this result set is also $\frac{r}{|Rel|}$. The explanation for R-p is taken from Chapter 8 of the book [39].

Hence, we need to devise some techniques that will work with different sets of output with different lengths, and consider some semantic knowledge-base from a standard ontology or thesaurus for comparison. The evaluation strategy should be strong enough to compare the systems and references against each-other (even it's own) performance and present them together in the same scale.

2.3. A Sound Baseline for All[‡]

We consider each document (and generated summary or keyphrases, given references, etc.) as a set of words (or phrases). We are flexible towards the set sizes except for the fact that the system generated summary or keyphrases and provided references are shorter than the original document. At first, in order to draw a base-case scenario, we assume that references and system outputs are complete subsets of the original document. Considering all these, we state hypothesis 1.

HYPOTHESIS 1. The size of overlap between two sets of output should be compared against the average intersection size of two random sets.

[‡] some parts of this section is reproduced from [14]; with permission from the Springer series

2.3.1. The Average Intersection Size

Let, We have a set N of size n, and two randomly selected subsets $K \subseteq N$ and $L \subseteq N$ with sizes k and l (say, $k \leq l$). The probability of any element x being present in both subset K and subset L is the probability that x is contained in the intersection of those two sets $I = L \cap K$.

$$\begin{aligned} \Pr(x \in K) \cdot \Pr(x \in L) &= \Pr(x \in (L \cap K)) \\ &= \Pr(x \in I) \end{aligned} \tag{1}$$

Putting another way, the probability that an element x is in K, L, or I is k/n, l/n and i/n respectively (where i is the number of elements in I). From equation 1 we deduce,

$$\begin{aligned} (k/n)(l/n) &= i/n \\ i &= \frac{kl}{n} \end{aligned} \tag{2}$$

A similar idea was stated briefly by Goldstein [11], but is not brought to much usage for evaluation purpose.

2.4. i-$measure$: A Length-independent Relativized Scale[§]

A direct comparison of an observed overlap (say, ω), seen as the intersection size of two sets K and L, consisting of lexical units like unigrams or n-grams drawn from a single set N is provided by the i-$measure$:

$$\begin{aligned} i\text{-}measure(N, K, L) &= \frac{observed\_size\_of\_intersection}{expected\_size\_of\_intersection} \\ &= \frac{|K \cap L|}{\frac{|K| \cdot |L|}{|N|}} = \frac{\omega}{\left(\frac{kl}{n}\right)} = \frac{\omega}{i} \end{aligned} \tag{3}$$

2.4.1. Connect Relativized Scale to Absolute Scale

Recall, Precision, and F-measure are the renowned absolute scales to define the performance of a system. Recall (r) is “the ratio of number of relevant information received to the total number of relevant information in the system”. Precision (p), on the other hand, is “the ratio of

[§] some parts of this section is reproduced from [14]; with permission from the Springer series number of relevant records retrieved to the total number (relevant and irrelevant) of records retrieved". Assuming the subset with size k as the gold standard, we define recall, and precision for the randomly generated sets as:

$$r = \frac{i}{k} \quad \text{and} \quad p = \frac{i}{l}$$

$$f\text{-}measure = \frac{2pr}{p+r}$$

$f\text{-}measure$ (the balanced harmonic mean of p and r) for these two random sets can be redefined using equation 2 as:

$$\begin{aligned} f\text{-}measure_{expected} &= 2pr/(p+r) \\ &= \tfrac{2\cdot i^2}{k\cdot l} / \tfrac{(k+l)\cdot i}{k\cdot l} \\ &= 2i/(k+l) \\ &= i/((l+k)/2) \end{aligned} \tag{4}$$

Let, for a machine-generated summary L and a reference summary K, the observed size of intersection, $|K \cap L|$ is ω.

$$r = \frac{|K \cap L|}{|K|} = \frac{\omega}{k} \quad \text{and} \quad p = \frac{|K \cap L|}{|L|} = \frac{\omega}{l}$$

$f\text{-}measure$, in this case, can be defined as,

$$\begin{aligned} f\text{-}measure_{observed} &= 2pr/(p+r) \\ &= \tfrac{2\cdot \omega^2}{k\cdot l} / \tfrac{(k+l)\cdot \omega}{k\cdot l} \\ &= 2\omega/(k+l) \\ &= \omega/((k+l)/2) \end{aligned} \tag{5}$$

By substituting ω and i using equation 5 and equation 4, we get,

$$i\text{-}measure(N, K, L) = \frac{f\text{-}measure_{observed}}{f\text{-}measure_{expected}} \tag{6}$$

Interestingly, $i\text{-}measure$ turned out as a ratio between the observed $f\text{-}measure$ and the expected/ average $f\text{-}measure$. In other words, "the $i\text{-}measure$ is a form of $f\text{-}measure$ with some tolerance towards the length of the summaries/keyword sets".

2.4.2. Adjust to the Length Variation

Suppose we have a document with $n = 200$ unique words, a reference summary composed of $k = 100$ unique words, and a set of machines $\{a, b, \ldots, h, i\}$. Each machine generates a summary with l unique words. Table 2.1 outlines some sample scenarios of $i\text{-}measure$ scores that would allow one to determine a comparative performance of each of the systems.

Table 2.1. A set of sample cases: $i\text{-}measure$

case	n	k	l	i	ω	$i\text{-}measure$	sys. id
	200	100	100	50	30	0.6	a
$k = l$	200	100	100	50	45	0.9	b
	200	100	100	50	14	0.28	c
	200	100	150	75	30	0.4	d
$k < l$	200	100	150	75	45	0.6	e
	200	100	150	75	14	0.186	f
	200	100	80	40	30	0.75	g
$k > l$	200	100	80	40	45	1.125	h
	200	100	80	40	14	0.35	i

For system b, e, and h, ω is the same, but the $i\text{-}measure$ is highest for h as its summary length is smaller than the other two. On the other hand, systems e and a receive the same $i\text{-}measure$. Although ω is larger for e, it is penalized as its summary length is larger than a. We can observe the following properties of the $i\text{-}measure$:

- The system's summary size (l) does not have to be exactly same as the reference' summary size size (k); which is a unique feature. Giving this flexibility encourages systems to produce more informative summaries.
- If k and l are equal, $i\text{-}measure$ follows the observed intersection, for example b wins over a and c. In this case i-measure shows a compatible behavior with recall based approaches.
- For two systems with different l values, but same intersection size, the one with smaller l wins (e.g. a,d, and g). It indicates that system g (in this case) was able to extract important information with greater compression ratio; this is compatible with the precision based

approaches.

2.4.3. Case Study1: Keyphrase Extraction

In order to explain the usefulness of the evaluation technique, we pick the keyphrase extraction task and a sample document from the dataset (Chapter 4) as our case study. We show why/how the relativized technique is more effective than the absolute scale.

Table 2.2. Author-written keyphrases and TextRank-generated keyphrases

Title	session-juggler secure web login from an untrusted ...
Author-assigned Keyphrases (K)	mobile, *session*, hijacking, secure, *login*, cookies
TextRank Generated Keyphrases (L)	session, user, site, phone, terminal, juggler, website, logout, browser, web, login, password, bookmarklet, http, untrusted

Table 2.2 shows that $|K| = 6$, $|L| = 15$, and $|N|$ was counted as 849. The average-size of intersection, $i = \frac{6*15}{849} = 0.016$. And there is 2 exact match, i.e. $\omega = 2$. Therefore, $i\text{-}measure = \frac{2}{0.016} = 18.866$. We can range the size of system output from 0 to a reasonable point, and find the corresponding i and ω, therefore, draw a performance graph for the system (e.g. figure 2.3).

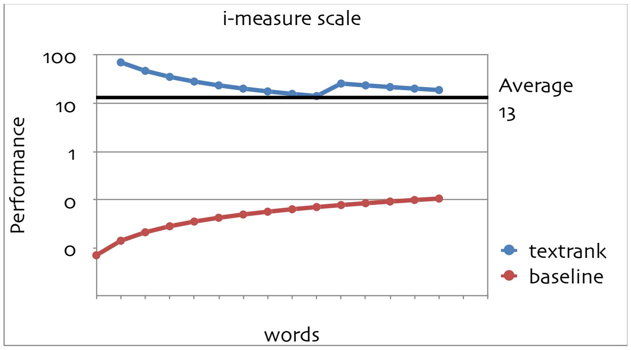


Figure 2.3. Performance of TextRank and Baseline in "Logarithmic" scale

We generate a series of $i\text{-}measure$ based points for a system by varying the extracted set size $l = \{l_1, l_2, ..., l_z\}$ upto some feasible point l_z. We plot them into a graph to predict the performance trend of the system. Now, using the same set of points (l), we can generate the corresponding i-measures of another system and plot it as another curve. From these two curves, we can predict or compare the performance variation or average performance between different systems. Our major concern is, different system-outputs should be tested several times against each-other in order to compare the performance. We plan to use some distance-based calculation (e.g. discrete fréchet distance, a way to determine the similarity between curves) to report the performance of the participants.

It is worth mentioning that several words from the TextRank output are closely related to the diamond standard: $\{$(login, logout), (mobile, phone), (login, password), (secure, untrusted), (mobile, user)$\}$ and so on. As we also see some TextRank-extracted words in the title, it is clear that these words are closely related and important for the paper. We should not completely ignore these phrases. Hence, it is necessary to have a new mechanism for evaluation. Using some domain specific ontology might be a better approach; but for now, we can at-least use the WordNet, which is quite large at size and usable for any domain. We will be discussing about the issue at Chapter 4.

2.4.4. Case Study2: Summarization

The example block shows the author written abstract, TextRank generated summary, and a summary produced by an anonymous algorithm for a particular scientific article. The anonymous algorithm basically extracts two sentences from the conclusion section. We apply the sentence tokenizer (from NLTK [32] package) and collect nouns, adjectives, verbs, and adverbs from each sentence. The original document is also processed through the same method.

aggregating crowdsourced binary ratings

abstract (K):

in this paper we analyze a crowdsourcing system consisting of a set of users and a set of binary choice questions. each user has an unknown, fixed, reliability that determines the user's error rate in answering questions. the problem is to determine the truth values of the questions solely based on the user answers. although this problem has been studied extensively, theoretical error bounds have been shown only for restricted settings when the graph between users and questions is either random or complete. in this paper we consider a general setting of the problem where the user question graph can be arbitrary. we obtain bounds on the error rate of our algorithm and show it is governed by the expansion of the graph. we demonstrate, using several synthetic and real datasets, that our algorithm outperforms the state of the art.

TextRank summary (L_1):

all users in a have reliability 1, those in b have reliability 1, and user x has reliability 0. we next show an error bound on the estimate w for user reliability obtained from algorithm 1. (user error bound). $error(w) <$ we evaluate both item rating estimates and user reliability estimates. next we analyze the error in user reliability estimates. we studied the problem of aggregating user ratings when the user item rating graph is arbitrary.

Anonymous algorithm (L_2):

We studied the problem of aggregating user ratings when the user item rating graph is arbitrary. We formulated a matrix completion problem and presented two eigenvector-based algorithms that have guaranteed error bounds when the resulting user-user co-rating graph satisfies expansion properties.

If we note the title and the abstract carefully, we see that some phrases (user, reliability, error rating, error bounds, arbitrary graph, crowdsource systems, graph expansion, etc.) are very significant. And most of them are present in both of system generated summaries. The observed intersection(ω) is same for both of the system outputs but the one with greater lengths get lower score over the other.

In both of the case studies, we present the way to calculate i-$measure$ with a single reference.

2.5. Summary of the Chapter

In this chapter, we have proposed a simple technique to define a generic baseline that compares the expected output considering three environment variables: number of words in the original document, number of words in the reference sample, and number of words in the system generated output. i-$measure$ is the direct comparison between the observed overlap and the expected overlap. As it adjusts to the length variation, we do not need to truncate the output at some fixed point;

Table 2.3. Performance of different systems

(a) Performance of TextRank(L_1) using i-$measure$

$K \cap L_1$	i-$measure$
reliability, problem,	$\|N\| = 818$
graph, error,	$\|K\| = 35$
arbitrary, algorithm,	$\|L_1\| = 15$
user, bound	$i = 0.6418$
	$\omega = 8$
	i-$measure = 12.464$

(b) Performance of an Anonymous algorithm (L_2) using i-$measure$

$K \cap L_2$	i-$measure$
graph, algorithm,	$\|N\| = 818$
rate, problem,	$\|K\| = 35$
expansion, error,	$\|L_2\| = 24$
user, bound	$i = 1.0268$
	$\omega = 8$
	i-$measure = 7.790$

or at least, we can be less restrictive towards the length and pay more attention to the qualitative features (fluency, syntactic well-formedness, informativeness) of the system output. Though we have used words as the unit of comparison, we can extend it to consider $phrases$ or $sentences$ as the evaluation unit.

Chapter 3 presents an i-$measure$ based evaluation technique using multiple references.

CHAPTER 3

IMPROVING EVALUATION WITH MULTIPLE REFERENCES[‡]

There is no one way to evaluate NLP systems, primarily because these are not autonomous entities: assuming that there is a version of the naturalistic fallacy which supposes that NLP aspires towards human LP capabilities without allowing for the fact that humans have different capabilities that are differently deployed in different circumstances [21].

3.1. Introduction

Human quality text summarization systems are difficult to design and even more difficult to evaluate [11]. The extractive summarization task has been most recently portrayed as ranking sentences based on their likelihood of being part of the summary and their salience. However different approaches are also being tried with the goal of making the ranking process more semantically meaningful, for example: using synonym-antonym relations between words, utilizing a semantic parser, relating words not only by their co-occurrence, but also by their semantic relatedness. Quite a few research groups are working on improving anaphora resolution techniques, defining dependency relations, etc. with a goal of improving the language understanding of a system.

A series of workshops on text summarization (WAS 2000-2002), special sessions in ACL, CoLING, SIGIR, and government sponsored evaluation efforts in United States (DUC 2001-DUC2007) have advanced the technology and produced a couple of experimental online systems [58]. However there are no common, convenient, and repeatable evaluation methods that can be easily applied to support system development and comparison among different summarization techniques [30]. Several studies ([65], [63], [41], [38]) suggest that multiple human gold-standard summaries would provide a better ground for comparison. Lin [27] states that multiple references tend to increase evaluation stability although human judgements only refer to single reference summary.

[‡] This Chapter is reproduced from [14]; with permission from Springer

After considering the evaluation procedures of ROUGE [28], Pyramid [52], and their variants e.g. ParaEval [74], we present another approach to evaluating the performance of a summarization system which works with one or many reference summaries. Our major contributions are:

- We propose the average or expected size of the intersection of two random generated summaries as a generic baseline (Chapter 2). Such a strategy was discussed briefly by Goldstein et al. [11]. However, to the best of our knowledge, we have found no direct use of the idea while scoring a summarization system. We use the baseline to find a related (normalized) score for each reference and machine-generated summaries.
- Using this baseline, we outline an approach (sec. 3.3) to evaluating a summary. Additionally, we outline the rationale for a new measure of summary quality, detail some experimental results and also give an alternate derivation of the average intersection calculation.

3.2. Related Work

At the early stages of developing robust evaluation strategies by Document Understanding Conference, a single model summary was used. System summaries were used to be evaluated manually. The average degree to which the model summary's clauses overlap with the system summary's content is called the coverage [33]. The coverage scores were used to indicate the quality of the system output. Later studies show that it is good to use multiple models for the evaluation purpose. As humans are prone to deviate from their own summary written at a different time, and multiple models add ambiguity, coverage scores can vary based on which model is considered.

The Pyramid evaluation [51] method has been developed for a reliable and diagnostic assessment of content selection quality in summarization. From several human models the annotators identify semantically defined units, named as Summary Content Units (SCUs). Each SCU is weighted based on the number of humans express that in their model summaries. An ideal maximal informative summary would express a subset of most highly weighted SCUs.

The original pyramid score is similar to a precision metric. It reflects the number of content units that were included in a summary under evaluation as highly weighted as possible and it penalizes the content unit when a more highly weighted one is available but not used. We would like to address following important aspects here -

- manually defined SCUs require another level of human labor besides collecting the gold standards.
- Pyramid method does not define a way to compare the degree of (dis)agreements between human models.
- High frequency SCUs receive higher weights in the Pyramid method. Nenkova [53], in another work, stated that the frequency feature is not adequate for capturing all the contents. To include less frequent (but more informative) content into machine summaries is still an open problem.
- The Pyramid score correlates well (Spearman's $\rho = 0.85$) while tested on TAC2009 Responsiveness task. But it is worth noticing that the responsiveness measure involves some linguistic quality whereas the Pyramid metric was designed to consider content only.
- There is no clear direction about the summary length (or compression ratio).

Manual evaluation, and detection of SCUs require significant human labor and comprehensive training. Therefore, DUC or TAC and individual research groups have adapted ROUGE [28] as their standard evaluation methodology. ROUGE (Recall Oriented Understudy for Gisting Evaluation) is closely modeled after BLEU [55], a package for machine translation evaluation. ROUGE automates the comparison between references and system outputs based on n-gram overlaps. So far, the overlap scores have been shown to correlate well with human assessment, thus surpassing the requirement of manual-labors for defining the SCUs.

Among the major variants of ROUGE measures, e.g. ROUGE-N, ROUGE-L, ROUGE-W, and, ROUGE-S, three have been used in the Document Understanding Conference (DUC) 2004, a large-scale summarization evaluation sponsored by NIST. Though ROUGE shown to correlate well with human judgements, it considers fragments of various lengths, to be equally important, a factor that rewards low informativeness fragments unfairly to relative high informativeness ones [17].

Nenkova [52] made two conclusions based on their observations:

- DUC scores cannot be used to distinguish a good human summarizer from a bad one
- The DUC method is not powerful enough to distinguish between systems

Considering the automatic overlap detection steps from ROUGE and weighed overlaps of SCUs, we propose our evaluation method that has the following properties:

- automatically detect n-gram chunks
- use $i\text{-}measure$ to relativize the output based on average random probability
- the normalized weighted relatedness of n-grams between models defines the importance of each model with respect to the other.

3.3. Confidence based Evaluation

When multiple reference summaries are available, a fair approach is to compare the machine summary with each of them. If there is a significant amount of disagreement among the reference (human) summaries, this should be reflected in the score of a machine-generated summary. Averaging the overlaps of machine summaries with human written ones does not weigh less informative summaries differently than more informative ones. Instead, the evaluation procedure should be modified so that it first compares the reference summaries among themselves in order to produce some weighting mechanism that provides a fair way to judge all the summaries and gives a unique measure to quantify the machine-generated summaries. In the following subsections we introduce the dataset, weighting mechanism for references, and finally, outline the scoring process.

3.3.1. Introduction to the Dataset and the System Structure

Our approach is generic and can be used for any summarization model that uses multiple reference summaries. We have used DUC-2004 structure as a model. We use $i\text{-}measure(d, x_j, x_k)$ to denote the i-measure calculated for a particular document d using the given summaries x_j and x_k.

Let λ machines ($S = \{s_1, s_2, \ldots, s_\lambda\}$) participate in a single document summarization task. For each document, m reference summaries ($H = \{h_1, h_2, \ldots, h_m\}$) are provided. We compute the $i\text{-}measure$ between $\binom{m}{2}$ pairs of reference summaries and normalize with respect to the best pair. We also compute the $i\text{-}measure$ for each machine-generated summary with respect to each reference summary and then normalize it. We call these normalized i-measures and denote them as

$$\begin{aligned} w_d(h_p, h_q) &= \tfrac{i\text{-}measure(d,h_p,h_q)}{\mu_d} \\ w_d(s_j, h_p) &= \tfrac{i\text{-}measure(d,s_j,h_p)}{\mu_{(d,h_p)}} \end{aligned} \tag{7}$$

where,

$$\begin{aligned} \mu_d &= max(i\text{-}measure(d, h_p, h_q)), \forall h_p \in H, h_q \in H, h_p \neq h_q \\ \mu_{(d,h_p)} &= max(i\text{-}measure(d, s, h_p)), \forall s \in S \end{aligned}$$

The next phase is to build a heterogeneous network of systems and references to represent the relationship.

3.3.2. Defining Confidence and $i\text{-}Score$

We assign each reference summary h_p a “confidence” $c_d(h_p)$ for document d by taking the average of its normalized $i\text{-}measure$ with respect to every other reference summary:

$$c_d(h_p) = \frac{\sum_{q=1,p\neq q}^{m}(w_d(h_p, h_q))}{m-1}. \tag{8}$$

Taking the confidence factor associated with each reference summary allows us to generate a score for s_j:

$$score(s_j, d) = \sum_{p=1}^{m} c_d(h_p) \times w_d(s_j, h_p) \tag{9}$$

Given t different tasks (single documents) for which there are reference and machine-generated summaries from the same sources, we can define the total performance of system s_j as

$$i\text{-}score(s_j) = \frac{\sum_{i=1}^{t} score(s_j, d_i)}{t}. \tag{10}$$

Table 3.1 shows four reference summaries (B, G, E, F) and three machine summaries $(31, 90, 6)$ for document D30053.APW19981213.0224. Table 3.2 shows the normalized $i\text{-}measure$ for each reference pair. While comparing the summaries, we ignored the stop-words and punctuations.

Table 3.1. Reference (B, G, E, F) summaries and Machine $(90, 6, 31)$ summaries

Reference	Summary
B	Clinton arrives in Israel, to go to Gaza, attempts to salvage Wye accord.
G	Mid-east Wye Accord off-track as Clintons visit; actions stalled, violence
E	President Clinton met Sunday with Prime Minister Netanyahu in Israel
F	Clinton meets Netanyahu, says peace only choice. Office of both shaky
90	ISRAELI FOREIGN MINISTER ARIEL SHARON TOLD REPORTERS DURING PICTURE-TAKIN=
6	VISIT PALESTINIAN U.S. President Clinton met to put Wye River peace accord
31	Clinton met Israeli Netanyahu put Wye accord

Table 3.2. Normalized $i\text{-}measure$ of all possible reference pairs

$Pair(p, q)$	n	k	l	ω	i	$i\text{-}measure$	$w_d(h_p, h_q)$
(G , F)	282	10	8	1	0.283687	3.525	0.375
(G, B)	282	10	9	3	0.319148	9.4	1.0
(G, E)	282	10	8	1	0.283687	3.525	0.375
(F, B)	282	8	9	1	0.255319	3.916	0.4166
(F, E)	282	8	8	2	0.226950	8.8125	0.9375
(E, B)	282	8	9	2	0.255319	7.8333	0.8333

Table 3.3. "Confidence Score"

reference: h_p	confidence: $c_d(h_p)$
G	0.583
F	0.576
B	0.75
E	0.715

System Id(s_j)	$score(s_j, d_i)$
31	0.2676
6	0.1850
90	0.0198

Table 3.4. Confidence-based "System Score"

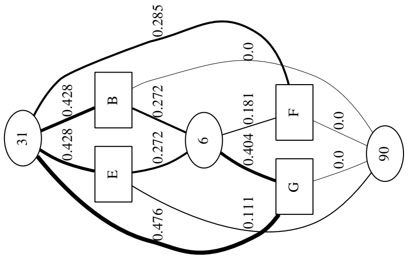


Figure 3.1. "System-Reference Graph": edge-weights represent the normalized $i\text{-}measure$

3.3.3. Evaluating Multi-document Summary

Methodology defined in Section 3.3.2 can be adapted for evaluating multi-document summaries with minor modifications. Let, there are q clusters of documents, i.e. $D = \{D_1, D_2, \ldots, D_q\}$. Each cluster D_i contains t number of documents, $D_i = \{d_1, \ldots, d_t\}$. The system has a set of humans ($H = \{h_1, h_2, \ldots, h_z\}$) to generate gold summaries. For each D_i, a subset of humans ($H_{D_i} = \{h_1, h_2, \ldots, h_m\}, m \leq z$) write m different multi-document summaries.

We need to compute a score for system s_j among λ participating systems ($S = \{s_1, s_2, \ldots, s_\lambda\}$). We, first, compute $score(s_j, D_i)$ for each D_i using formula 9. Then we use formula 10 to find the rank of s_j among all participants.

The only difference is at defining the $i\text{-}measure$. The value of n (total number of units like unigram, bi-gram etc.) comes from all the participating documents in D_i, other than a single document.

3.4. Experimental Results

We perform different experiments over the dataset. Section 3.4.1 describes how $i\text{-}measure$ among the reference summaries can be used to find the confidence of judgements. In Section 3.4.2, we examine two types of rank-correlations (pair-based, distance based) generated by $i\text{-}score$ and $ROUGE$-1. Section 3.4.3 states the correlation of $i\text{-}measure$ based ranks with human assessors.

3.4.1. Correlation between Reference Summaries

The $i\text{-}measure$ works as a preliminary way to address some intuitive decisions. We discuss them in this section with two extreme cases.

- If the $i\text{-}measure$ is too low (table 3.6) for most of the pairs, some of the following issues might be true:-
  - The document discusses about diverse topics.
  - The compression ratio of the summary is too high even for a human to cover all the relevant topics discussed in the document.
  - The probability of showing high performance by a system is fairly low in this case.
- If the $i\text{-}measure$ is fairly close among most of the human pairs (table 3.2), it indicates:-

- The compression ratio is adequate
- The document is focused into some specific topic.
- If a system shows good performance for this document, it is highly probable that the system is built on good techniques.

Therefore, the $i\text{-}measure$ could be an easy technique to select ideal documents that are good candidates for summarization task. For example, table 3.2 shows that all of the reference pairs have some words in common, hence their confidence score(table 3.3) is fairly high. But table 3.7 shows that most of the references do not share common words, hence confidence values of the references for document $D30015.APW19981005.1082$ (Table 3.5) is quite different from each other.

Table 3.5. Another sample of "Reference" summaries

Reference	Summary
A	U.S. envoy Holbrooke gives last minute warning to Milosevic on Kosovo
B	US envoy tells Milosevic, stop Kosovo crackdown or face NATO airstrikes.
H	Clinton, Cook, NATO military action, Yeltsin, Sept. 23 U.N. resolution
E	Yugoslavia declares UN warnings about Albania are a "criminal act."

3.4.2. Correlation of Ranks: ROUGE-1 vs. i-Score

To understand how the confidence based i-measures compare to the ROUGE-1 metric, we calculated Spearman's ρ [66] and Kendall's τ [24], (both of which are rank correlation coefficients) by ranking the machine and reference summary scores. Spearman's ρ considers the squared difference between two rankings while Kendall's τ is based on the number of concordant and discordant pairs (Table 3.8). Since the list of stopwords used by us can be different from the one used by ROUGE system, we also calculate pure $f\text{-}measure$ based rank and report the correlation of with $i\text{-}score$. The results show, for both cases, i-measure is positively correlated, but not completely.

Table 3.6. Another sample of normalized $i\text{-}measure$ of all possible reference pairs

$Pair(p,q)$	n	k	l	ω	i	$i\text{-}measure$	$w_d(h_p,h_q)$
(A, H)	357	9	10	0	0.25210	0.0	0.0
(A, B)	357	9	10	3	0.25210	11.9	1.0
(A, E)	357	9	7	1	0.17647	5.66	0.4761
(H, B)	357	10	10	1	0.2801	3.57	0.3
(H, E)	357	10	7	0	0.19607	0.0	0.0
(B, E)	357	10	7	0	0.19607	0.0	0.0

Table 3.7. "Confidence Score" of each reference

reference: h_p	confidence: $c_d(h_p)$
A	0.492
B	0.433
H	0.099
E	0.158

3.4.3. Correlation with Human Judgement: Guess the RESPONSIVENESS score

For multi-document summarization (DUC2004, task5), the special task (responsiveness) was to assess the machine summaries per cluster (say, D_i) by a single human-assessor (h_a) and score between 0 to 4, to reflect the responsiveness on a given topic (question). We have used a histogram to divide the $i\text{-}score$ based space into 5 categories ($\{0,1,2,3,4\}$). We found 341 decisions out of 947 responsiveness scores as an exact match (36.008% accuracy) to the human assessor. Table 3.9 is a snapshot of the scenario.

The Root Mean Square Error (RMSE) based on $i\text{-}score$ is 1.212 at the scale of 0 to 4. Once normalized over the scale, the error is 0.303

$$RMSE = \sqrt{1/n \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$

3.4.4. Endorsing Syntactic Well-formedness

After carefully analyzing the system generated summaries, rouge based scores, and i-score, we noticed that most of the systems are not producing well-formed sentences. Scoring based on weighted or un-weighted overlapping of bag of important phrases is not the best way to evaluate a summarizer. Constraint on the length of the summary (byte or word) might be a trigger. As

Table 3.8. Rank correlation between $ROUGE$ and $i\text{-}measure$

i-score vs. ROUGE-1	Spearman's ρ	Kendall's τ
Task 1	0.786	0.638
Task 2	0.713	0.601
Task 5	0.720	0.579
i-score vs. f-measure		
Task 1	0.896	0.758
Task 2	0.955	0.838
Task 5	0.907	0.772

Table 3.9. "Guess-Score" for $D188$, assessor F

sys. id	$given\_score$	$guess\_score$
147	3	2
43	2	3
71	2	0
122	2	2
B	4	4
86	2	0
24	1	1
49	2	1
116	1	1
109	3	3
H	3	4
F	4	4

$i\text{-}measure$ is lenient on lengths, we can modify equation 9 with the following to apply extraction or generation of proper sentences within a maximum word or sentence window as an impact factor.

$$score(s_j, d) = \left(\sum_{p=1}^{m} c_d(h_p) \times w_d(s_j, h_p)\right) \times \frac{c\_sen}{t\_sen} \tag{11}$$

where, t_sen is the total number of sentences produced or extracted by s_j and c_sen is the number of grammatically well-formed sentences. For example,"This is a meaningful sentence. It can be defined using english grammar." is a delivered summary. Suppose, the allowed word-window-size is 8. So the output is chopped as "This is a meaningful sentence. It can be". Now it contains 1 well-formed sentence out of 2. Then the bag of words or phrases model (e.g. $i\text{-}measure$) can be applied over it and a score can be produced using equation 11.

Standard sentence tokenizers, POS taggers, etc. can be used to analyze sentences. The word or sentence window-size can be determined by some ratio of sentences (words) present in the original document. As we could not find any summary-evaluation conferences who follow similar rules (TREC, DUC, etc.), we were unable to generate results based on this hypothesis.

3.5. Summary of the Chapter

We present a mathematical model for defining a generic baseline. We also propose a new approach to evaluate machine-generated summaries with respect to multiple reference summaries, all normalized with the baseline. The experiments show comparable results with existing evaluation techniques (e.g. ROUGE). Our model correlates well with human decision as well.

The $i\text{-}measure$ based approach shows some flexibility with summary length. Instead of using average overlapping of words/phrases, we define pair based confidence calculation between each reference. Finally, we propose an extension of the model to evaluate the quality of a summary by combining the bag-of-words like model to accredit sentence structure while scoring.

We will be extending the model, in future, so it works with semantic relations (e.g. synonym, hypernym etc.) We also need to investigate some more on the confidence defining approach for question-based or domain-specific summary evaluation task.

Chapter 4 shows an extension of the $i\text{-}measure$ that considers some semantic information while evaluating the system outputs. It also presents a dataset that can be used by others for summary generation and keyphrase extraction tasks.

CHAPTER 4

BUILDING THE “DIAMOND STANDARD” DATASET

Summarization research is notorious for its lack of adequate corpora, a situation that prevents rapid progress in the field: today there exists only a few small collections of texts whose units have been manually annotated for textual importance [42].

4.1. Introduction

Automatic keyphrase extraction and summarization have been proved very useful for different applications. Given today’s very large collection of documents, these tasks are extremely important for the search and retrieval of relevant information. A reliable dataset is extremely helpful to evaluate the systems performing these tasks. The dominant evaluation approaches (ROUGE, Pyramid, etc.) need some references/models (known as gold-standards) to compute the performance of the systems/algorithms. To develop “gold-standards” manually on a large scale dataset is time-consuming and expensive, as well as subjective - differences between summaries (or keyphrases) written by different people can be significant.

Our effort in this phase is to publish a moderately large collection of pre-processed scientific articles with a standard set of references so that researchers can use it to train/test their algorithms. At the same time, we present a modified version of evaluation methodology (Chapter 2) that not only can adjust to the length variation between reference set(s) and the system generated outputs, but also defines an “equivalence class” representing each word in order to reduce string based mismatch effect.

Texts are not simply a flat list of sentences; they have a hierarchical structure, one in which certain clauses are more important than the others [16]. The statement is more appropriate for scientific articles. Also, the scientific articles come with author-assigned abstracts and keyphrases who can be used as a set of high quality references. They can also be treated as the baseline for the evaluation of system generated outputs. So, we call this collection of abstracts and keyphrases a “diamond standard”. We preprocess a set of scientific articles (excluding abstract, title, and reference list) to build the dataset. In order to test the usability of the dataset and the diamond-standard, we use the classic TextRank algorithm as the state of the art. Since the diamond standard is abstractive by nature, and can vary over length, we use a relativized-scoring mechanism (a variant of $i\text{-}measure$) to evaluate computer generated summaries and keywords. We provide the evaluation tool along with the dataset so that other systems can compare their performances with the state of the art.

We believe, a set of published papers (scientific articles) from which we extract the author-provided summary and keywords is an ideal baseline for both of the tasks. Our data-set contains more than 1000 published articles from different WWW and KDD proceedings. As scientific articles revolve around domain-specific ideas, the quality of the summary largely depends on the summarizer's knowledge (or, ability) to understand the topic of the article. For example, it is not a good idea to have one's favorite category-theory, genomics, or string-theory paper summarized by the non-expert evaluators from Mechanical Turk. The authors, in such cases, are the most suitable (or trustworthy) persons to summarize the article. Hence we automatically parse the abstract as a reference summary and the author-assigned keyphrases as reference keyphrases. We call the automatically parsed collection of references a diamond standard.

Besides parsing/cleaning the collection, we address another issue. Human annotated references are mostly abstractive; whereas system produced ones are extractive. For example, out of our 1250 collected documents, 1218 of the abstracts contain some words (excluding stopwords and punctuations) which are not present in the original document. And, 1186 of them contain words in the author-provided keyphrases that cannot be found in the original article. So, some form of semantic equivalence relation is needed, instead of exact word, phrase, or string matching strategy.

The references and the generated summaries and keyword sets might also have different lengths. Hence we propose to change our evaluation techniques from the traditional approach (precision, recall, f-measure) to a relativized approach, e.g. $i\text{-}measure$ [14]. In order to show the usage of our dataset, we produce some candidate-output using the classic unsupervised algorithm: TextRank [45]. We also provide TextRank's performance statistics on the relativized scale and a tool that can be reused by other candidate systems to evaluate their performances.

4.2. Available Datasets

Keyphrases (and summaries) need to be gleaned from the details of the documents. Several efforts are going on to create reliably larger datasets for the tasks. A few groups of researchers have published some qualitative datasets for keyphrase extraction. Inspec dataset [18] comes with 2000 abstracts as an attempt to enhance the performance of automatic keyword extraction. The SP dataset [54] contains around 120 scientific articles with author-assigned and reader-assigned keyphrases. The DUC dataset [67] contains 308 documents from DUC2001, manually annotated by two indexers. Task 5 of SemEval 2010 [25] published the ranks of 19 systems performing automatic keyphrase extraction from 284 compiled scientific articles.

On the other hand, the Document Understanding Conference (DUC) and the Text Analysis Conference (TAC) are two series of evaluation workshops that provide a large test collection, common evaluation procedures, and a forum for organizations to share their results for the summarization tasks. DUC provides some standard unstructured text data sets which the NLP community uses for evaluating summarization systems. The TIPSTER Text Summarization Evaluation Conference (SUMMAC) published a corpus of 183 documents from the Computation and Language collection. The ACL Anthology Network (AAN) [61] is a manually curated networked database of citations, collaborations, and summaries in the field of Computational Linguistics.

However, there are no common, convenient, and repeatable evaluation methods that can be easily applied to support system development and comparison among different summarization (or keyphrase extraction) techniques [30]. Besides, it is time-consuming and very expensive to produce human annotated reference set(s) for a large collection.

4.3. Building the Dataset

We have collected data (conference papers) from several proceedings of WWW and KDD using $CiteSeer^x$ digital library. Our choice for WWW and KDD was motivated by the availability of author-assigned keyphrases for each paper. We avoided posters as their internal structure is more complex and contains more figures than texts. Our dataset is composed with a total of 1000 articles. We have used the "pdf2txt" tool in order to convert the original paper to a plain text format. We applied several techniques such as POS-tagger, and an online dictionary to detect/rejoin common words that were split due to the complex pdf structure or column division. The dataset is categorized into three sections:

- diamond standard dataset (original paper and it's raw text, body of the paper, abstract, keyphrases)
- classic TextRank generated output (summary, keyphrases, $i\text{-}measure$ based statistics)
- evaluation tool (python source code)

The dataset can be downloaded from:

`https://github.com/FahmidaHamid/dataset_ictir`

4.3.1. Dataset@GitHub

This repository contains the following folders:

- "wwwSampleDataSet": original pdf files
- "wwwSampleDataSetOut": raw text files from the corresponding pdfs
- "www_ABS": abstract of each file (diamond-standard summary)
- "www_KE": author-written keyphrases of each file (diamond-standard keyphrases)
- "www_No_ABS": clean text extracted from "wwwSampleDataSetOut", (title, abstract, kephrases, section-names, and references are excluded)
- "www_Title": the title of each paper
- "www_References": the references (raw text) of each paper

We intend to use files at "www_No_ABS" as the given document for both purposes (summary generation, and keyphrase extraction). A few evaluation conferences, and most of the recent works who produce keyphrases from scientific articles tend to use only the abstract as the given text. We believe, an abstract is not large enough for a good algorithm to cover all possible keyphrases that it can, or it is supposed to extract or generate.

4.4. Evaluation Strategies So Far

Accurate computer-based evaluation of system-generated summaries (or keyphrases) is far from a being obvious or easy. Most of the shortcomings might come from the simplifications that statistical measures need to assume. The existing evaluation approaches use absolute scales (e.g. precision, recall, f-measure) to evaluate the performance of the participating systems. Such measures can be used to compare summarization algorithms, but they do not indicate how significant the improvement of one summarizer over another is [11]. ROUGE [28], the most re-known evaluation methodology, includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. Another summary evaluation tool Pyramid [51] considers multiple models to build a gold standard for system output. One drawback of both of these approaches are they are designed based on the availability of multiple gold standards per sample. We have discussed about them in detail at Chapter 3 and Chapter 2.

The evaluation of keyphrase extraction has received much less research attention than summarization. In most of the cases, researchers use the average of precision metric to compare their system's performance with baselines and other systems. Also, clarifying the baseline for a keyphrase extraction task is not well defined at all. If multiple references are available, there is no proper definition of how to weigh the agreement or disagreement between them. Another important issue is, same as summarization task, the length of reference set is not necessarily the same as the system generated sets. R-p [72] (a deviation from pure precision) was proposed by authors to consider various lengths of outputs together, but there is no way to adjust to the variation of length.

Considering these approaches, we can state that the absolute scales suffer due to the length constraints. We believe, comparing two summaries (or two sets of keyphrases) derived from the same source is sensitive to their lengths and the length of the source document they are extracted from. We, therefore, will use the relativized scale (discussed at Chapter 2) with some weighted-matching strategy that is suitable for evaluating both of the tasks.

While evaluating with an absolute scale, we control or trim the length of the output according to the reference's length to get a fair scenario. Then, we either use an exact phrase/word matching technique or have to employ manual labor for generating paraphrases. We propose a modified approach of i-$measure$ [14] that not only can adjust to the length variation but also considers an equivalence-relations using WordNet provided synsets to modify the observed intersection (matching) size.

4.5. Adapting Abstractive-ness

After exploring the existing automatic summarization and keyphrase extraction algorithms, we have discovered that the most of the algorithms focus on extractive methods. The majority of the reference sets, on the contrary, are abstractive by nature. When a human is asked to produce a summary (or write a set of representative phrases), he often produces new sentences/keyphrases, some of them possibly not occurring as such in the document. Therefore, our hypothesis, that the reference-set is a complete subset of the article ($K \subseteq N$, assumed in Section 2.3.1) is not always true. Similar statement also holds ($L \not\subseteq N$) if the algorithm uses abstractive summarization or keyphrase-extraction strategy.

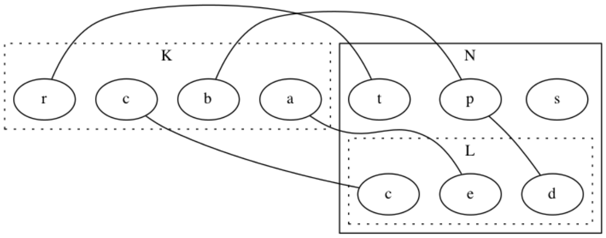


Figure 4.1. Words in N, L, and K are related

Suppose, word b is a hypernym of word p; and word d is a synonym of p (Figure 4.1). So they are related. If we use exact string matching strategy, we will miss the information present at the reference K and also, the machine-generated output will not be evaluated appropriately. Through the example, we verify the importance of covering semantic relatedness by the evaluation approach.

In such circumstances, we should deviate from the exact string or phrase matching approach to a weighted relatedness approach. To this end, we relate the basic units (words, in this case) found in the system generated outputs to the units in the reference sets. To be meaningful, the semantic relations need to be relativized to an ontology. As the first approximation, we use WordNet to find synset-based similarity for evaluation; a richer, more knowledge-intensive source like Wikipedia[*] is planned as a future development. Table 4.1 shows an example of synset-based comparison.

[*] `https://www.wikipedia.org/`

4.5.1. The Equivalence Class

WordNet [48] groups words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can be seen as a combination of dictionary and thesaurus. Both nouns and verbs are organized into hierarchies, defined by hypernyms or IS-A relationship. The words at the same level represent synset members. Each set of synonyms has a unique index, representing an equivalence class. We will test the overlap between two equivalence classes to count a match, along with the exact string or word matching approach.

Table 4.1. Synset-based "Equivalence Class" where $K \in abstract$ and $L \in TextRank$

K	Synset(K)	Synset(L)	L
data	{('datum.n.01'), ('data.n.01')}	{('information.n.02'), ('data.n.01'), ...}	information
aspect	{('expression.n.01'), ('aspect.n.04'), ('view.n.02'), ...}	{('view.n.03'), ('opinion.n.01'), ..., ('view.n.02')}	view
case	{ ('lawsuit.n.01'), ('case.n.01'), ('event.n.02'), ...}	{('model.n.07'), ('exemplar.n.01'),('example.n.01'), ('case.n.01'), ...}	example
shape	{('form.n.07'), ('condition.n.05'), ('shape.n.01'),...}	{('contour.n.03'), ('shape.n.01'), ('contour.n.01')}	contour
novel	{('fresh.s.04'), ('novel.s.02')}	{('new.s.04'), ('modern.s.05'), ('fresh.s.04')...}	new

We have applied a POS-tagger to fine-tune the synsets for summaries. While evaluating keyphrases, we have considered only words POS-tagged as $nouns$.

4.5.2. Word Embedding Vectors: A Supplement of WordNet

Recently deep learning based word embedding vectors have drawn the attention among researchers. Distributed representation of words in a vector space help learning algorithms to achieve better performance in natural language processing tasks by grouping similar words [47]. Through our research, we have been exploring some well established word embedding vectors to see how much information we can feed to the system and co-ordinate them in order to leverage the performance. For each word found in the original document, we wanted to add top five similar words to the text graph. We have calculated the cosine similarity between each pair of words from the embedding dictionary (SENNA [6], HLBL [50], dependency based vectors [26], etc.). It is important to note that these vectors were mostly generated from Wikipedia, Brown Corpus[†], etc.

[†] https://en.wikipedia.org/wiki/Brown_Corpus

As they were developed from a domain-independent database, they could not provide interesting results for scientific terms. It turned out that a set of word vectors have to be trained using a dataset of similar domain as the test dataset in order to get better results. In conclusion, we would like to state that word embedding vectors can be defined or trained automatically, but they are domain-dependent and a large computation power is required to train such model. Hence our attempt to use them as a supplement of WordNet for finding the equivalence class has not been very successful.

4.6. Scoring Approach

For the scientific article set, we combine the diamond standard consisting of author-provided summaries and keywords together with TextRank as a test system. For each paper, the TextRank algorithm generates a summary and a set of keywords. The TextRank graph builds directed edges between the words if they co-occur within a window range and/or they share some common synsets.

After generating keyphrases and summaries, we use a modified version of equation 3 to score the system. We adapted the WordNet provided synsets to detect similarity between two synsets of the system output and the diamond standard. The evaluation code considers only one highly trustable ("diamond standard") reference per document contrary to systems like ROUGE or confidence-based Scoring [14]. As a reference, we can look back at Section 2.4.3 to understand the importance of introducing an equivalence class while comparison. The code can be downloaded from:
`https://github.com/FahmidaHamid/code_ictir`.

4.7. Code Repository@ GitHub

The results are written in www_result_stat directory. We have two evaluation files in it. Each line of both of the files contains the following information:

- $file\_name$: name of the file
- n : total number of words in the file (excluding stopwords)
- k: total number of words in the reference (abstract/keyphrases)
- l: total number of words in the system-output (abstract/keyphrases)

- ω: number of words found in set $(K \cap L)$
- $partial\_omega$: number of words in $|K \cap L| - \omega$, considering the synsets
- $rand\_avg$: average size of intersection based on k and l (baseline)
- $i\_measure\_o$: score considering ω and $rand\_avg$
- $i\_measure\_p$: score considering $\omega + partial\_omega$ and $rand\_avg$

The repository also contains our implementation of the TextRank algorithm (TextRank.py) that uses WordNet based synsets to connect words while creating the textgraph. Also, parseDocument.py tells us how we have extracted information from the raw_text, cleaned it, re-formed some words with the help of a dictionary, and prepared the dataset (Section 4.3.1).

The most important one is evaluate_i_measure.py. We have used python, version 3 to prepare the code. Several other packages (NLTK, WordNet, etc.) will be required to execute it. “trPathSumm” and “trPathKE” are set to the locations where the system generated outputs for the summary and the keywords are stored respectively. If one wishes to go with just one of the two, the other path can be left empty. The class-object “evaluate_i_measure” has two boolean parameters to indicate which of the tasks (summary-generation, keyphrase-extraction) are required to be evaluated. Both of them are set to $True$ state at the provided code. But either one can be turned off by replacing its value with $False$ while calling the “iterOverAll” function. Once all the required packages are installed, and the repository and system output paths are set to appropriate values, a simple command like

$$python3\ evaluate\_i\_measure.py$$

will produce two files (ke_stat.dat, summ_stat.dat) at www_result_stat directory. On the contrary, only summ_stat.dat file will be created if the boolean parameters are set as $(True, False)$ while calling function “iterOverAll”.

4.8. Summary of the Chapter

Accurate evaluation is useful - including for their use in machine learning. Tools like the *i-measure* introduce some flexibility. Our version of *i-measure* is tolerant towards length variation and uses some semantic information while calculating the system’s performance. Besides

WordNet, the tool can be enriched with some domain-specific ontologies (Gene, FOAF, GoodRelations, YAGO, etc.) and paraphrase detection techniques. The dataset, we publish here, is large, automatically cleaned, and usable for both keyphrase extraction and summarization tasks. It comes with a trustable set of reference summary and keyphrases. Unsupervised algorithms can also use it to test their performances. We believe, it will be a great resource to the research community who are working with information extraction from scientific articles.

In Chapter 5, we will be introducing a case study to generate sentiment-oriented summary as part of the evaluation process. This is an example of extrinsic summarization which aims to generate summaries to serve a specific purpose (maximize the presence of sentiment in the summary). We compare it with TextRank as a state-of-the-art technique, and use the mathematically sound baseline to test the informativeness of the sentiment-oriented summaries.

CHAPTER 5

A CASE STUDY: SENTIMENT-ORIENTED SUMMARY GENERATION AND EVALUATION[†]

5.1. Introduction

A document expresses a writer's opinions along with some facts. Usually an article covering several issues will qualify some with positive feedback and some with negative. A high quality summary should reflect the most "important" ones among them. Summarization is thus closely related to sentiment analysis. There has been limited work done on the intersection of text summarization and sentiment analysis. Balahur [1] showed a technique of sentiment-based summarization on multiple documents. They have used a supervised sentiment classifier to classify the blog sentences into three categories (positive, negative, and objective). The positive and the negative sentences are, then fed to the summarizer separately to produce one summary for the positive posts and another one for the negative posts. The success of their model mostly depends on the performance of the sentiment classifier. It is important to note that their summarizer does not consider the impact of positive (negative) sentiments while producing the summary of negative (positive) sentiments. It sounds unintuitive to totally separate the sentiment-flows before producing the summaries. Manning [2], in their sentiment summary paper used Rotten Tomatoes dataset (for training and testing) to extract the most important paragraph from the reviewer's article. They aimed at capturing the key aspects of author's opinion about the subject (movie). They worked with a supervised technique and articles with single topic.

In this work, we propose an unsupervised, mutually recursive model that can represent text as a graph labeled with polarity annotations. Our model builds a graph by collecting words, and their lexical relationships from the document. It handles two properties ("bias" and "rank") for each of the important words. We consider sentiment-polarity of words to define the bias. The lexical definition and semantic interactions of one word to others help defining edges of the text-graph. We, thus, build a weighted directed graph and apply our model to get the top (positively

[†] This Chapter is reproduced from [15]; with permission from Springer

and negatively ranked) words. Each word in our graph starts with the same rank, which eventually converges to some distinct values with the effect of bias of its neighbors and weighted in-links. Those words then specify the weight of each sentence and grant us a direction to choose important ones. The bias of a node gets updated from the rank of it's neighbors. The mutual dependency of the graph elements represents the impact of the author's sentiment. Our concept is analogous to TextRank algorithm, except -

- Our model works for a polar graph whereas TextRank works with non-polar one.
- The rank of a node in TextRank gets updated by the connectivity (weighted or unweighted), whereas the rank in our model gets updated based on the weighted links and bias of its neighbors.

To the best of our knowledge, our concept of "anti-summary" (constructed with negatively polarized sentences) is new. Hence, it was hard to compare the results with a gold standard. We have chosen DUC2004 dataset and basic TextRank algorithm for comparative study. Through our experiments, we have found the following interesting facts -

- When the anti-summary and summary are mostly disjoint, the document is a collection of different sentiments on stated topics. It can be a transcript from some debate, political talk, controversial news, etc.
- When the anti-summary overlaps at a noticeable amount with the summary, the document is a news article stated from a neutral point of view.
- By blending anti-summary with TextRank generated one, we show another way of producing opinion-oriented summary which not only contains the flow of negative sentiment but also includes facts (or some positive sentiment).

5.2. Related Work

Automated text summarization dates back to the end of fifties [35]. A summarizer deals with the several challenges. To extract important information from a huge quantity of data while maintaining quality are two of them. A summarizer should be able to understand, interpret, abstract, and generate a new document. Majority of the works focus on "summarization by text-span extraction" which transforms the summarization task to a simpler one: ranking sentences from the original document according to their salience or their likelihood of being part of a summary [11].

Early research on extractive summarization was based on simple heuristic features of the sentences such as their position in the text, frequency of words they contain etc. More advanced techniques also consider the relation between sentences or the discourse structure by using synonyms of the words or anaphora resolution. To improve generic machine generated summaries, some researchers [11] converted some hand-written summaries (collected from the Reuters and the LosAngeles Times) to their corresponding extracted ones. Based on their experiments, they stated that summary length is independent of document length. Though Hovy and Lin [16] stated earlier, "A summary is a text that is produced out of one or more texts, that contains the same information of the original text, and that is no longer than half of the original text." For our experiments, we will generate summaries with at-most top ten sentences per document.

Graph based ranking algorithms have recently gained popularity in various natural language processing applications; specially in generating extractive summaries, selecting keywords, forming word clusters for sense disambiguation, and so on. They are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph [45]. The basic idea is of "voting" or "recommendation". When one vertex links to another one, it is basically casting a vote for the other vertex. The importance of the vertex casting the vote determines how important the vote itself is. Hence the score (usually called "rank") associated with a vertex is determined by the votes cast for it, and the score of the vertices casting these votes. TextRank works well because it does not rely on the local context of a text unit and requires no training corpus, which makes it easily adaptable to other languages or domains. Erkan and Radev [7] in LexRank (another graph based ranking algorithm to produce multi-document summary) used the centrality of each sentence in a cluster to assess the importance of each sentence. To measure the similarity between two sentences, they used cosine similarity matrix (based on word overlap and idf (inverse document frequency) weighting). Being inspired by the success of TextRank models, we had the idea to apply a polar-TextRank model in order to extract sentences from negative (positive) point of view.

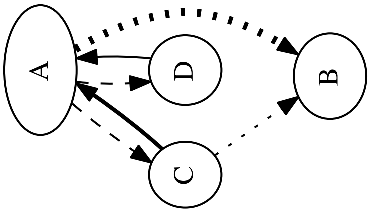


Figure 5.1. A "Text-Graph" describing several "Topics"

It is important that we consider each sentiment of the author while producing the summary. In our work, we adopt a graph based ranking model which originally was proposed for trust-based (social, peer-to-peer) networks [49]. It intends to measure the prestige (rank) of nodes (participants in the event) present in the graph. Their hypothesis, "the node which is prone to trust (mistrust) all its neighbors is less reliable than the node who provides unpredictable judgments," works also for producing summaries. Each node (word) has weighted (positive/negative/neutral) directed links to its neighbor nodes (other words, possibly collected from the same sentence or nearby sentences). The more weight it provides to its neighbors the more importance (either positive or negative) it indicates. The impact is higher when a node behaves differently (a positive biased node has a negative weighted outline or vice versa) towards its neighbors.

5.3. Anti-Summary

We propose an extractive summarization technique which produces anti-summaries as well as summaries for each document. We would discuss what anti-summary is and why it is important. Sentences with upstream knowledge are the candidates of anti-summary. A sentence does not have to contain words like $\{$no, neither, never, not, ever, bother, yet, $\ldots\}$, to be the part of the anti-summary.

We can start with a generic example: A document is about topic A. It is comparing the qualities of A with related topics B, C, and D. Suppose, topic B is mostly receiving negative opinions in that document. Then a summary of the document should include positive feedbacks on A and the anti-summary should be more about the properties of B.

Anti-summaries are as important as summaries. They help us find relative materials on a specific topic. For example, from a news article, without any supervised topic detection, anti-summaries can indicate which parts of it represent negative/ suppressed opinion. In a scientific article, anti-summaries tell how system A is different, better, or worse than system B where as summaries might only tell us the usefulness of system A.

Interestingly enough, some summary sentences are also present in anti-summary of the document. This means, an anti-summary is not exactly opposite to a summary, it is the reverse stream of the main news. Anti-summaries can help a search engine build comparative database. It is intuitive that two documents are related if there is a significant match between one's summary and the other's anti-summary.

5.4. Sentiment Analysis: Covering Minimal Issues

Sentiment Analysis has important aspect on fields which are affected by people's opinions, e.g. politics, economics, social science, management science and so on. It plays a vital role in every aspect of NLP; for example, co-reference resolution, negation handling, word sense disambiguation etc. Sentiment words are instrumental to sentiment analysis [31]. Words like good, wonderful, amazing convey positive connotation whereas bad, poor, terrible are used to flow negative sense. As an exception, some adjectives and adverbs (e.g. super, highly) are not oriented clearly to either one of the poles (positive, negative). A list of sentiment words are called sentiment lexicon. A sentiment lexicon is necessary but not sufficient for sentiment analysis.

A positive or negative sentiment word may have opposite orientations in different application domains, e.g. "The vacuum cleaner sucks!" vs. "The camera sucks." Sentiment words may be used objectively rather subjectively in some sentences, e.g. "If I can find a good camera, I can buy that." Sarcastic sentences are trickier to handle, as well as sentences having no sentiment word but with a sentiment expressed.

Based on the level of granularities (document level, sentence level, entity and aspect level) we choose the entity level analysis of sentiments. For example, the sentence, "The iPhone's call quality is good, but its battery life is short," evaluates two aspects: call quality and battery life. The two opinion targets for this sentence, call quality has positive sentiment and battery life has negative.

Our model is unsupervised, and we decided not to use statistical database to calculate the sentiment polarity for sentences/ paragraphs/ document. Hence we have used only a sentiment lexicon to get the usual sentiment polarity at word level.

5.5. The Polarity based TextRank Model

Jon Kleinberg's HITS or Google's PageRank are two most popular graph based ranking algorithms, successfully used for analyzing the link-structure of world wide web, citation graph, and social networks. A similar line of thinking is applied to semantic graphs from natural language documents, resulting in a graph based ranking model, TextRank [45]. The underlying hypothesis of TextRank is that in a cohesive text fragment, related text units tend to form a web of connections that approximates the model humans build about a given context in the process of discourse understanding. TextRank, with different forms (weighted, unweighted, directed, undirected) of graphs, was applied successfully for different applications, specifically for text summarization [46]. Based on the results so far, it performed well for summarizing general text documents. There are documents which present arguments, debates, competitive results and they are subjective reflections of the author(s). The limitation of TextRank (and other similar models) is that it does not handle negative recommendations different from positive ones. In this work, we present a different model [49] that can be adopted to have the impact of sentiments on the summary.

5.5.1. Trust-based Network

A network based on trust (e.g. facebook, youtube, twitter, blogs) is quite different from other networks; in each case, reputation of a peer as well as types of opinion (trust, mistrust, neutral) matters. An explicit link in a trust-based network indicates that two nodes are close (connected), but the link may show either trust or mistrust. A neutral opinion in a trust-based network is different from no-connection. TextRank gives higher ranks with better connectivity. The situation changes dramatically in trust-based networks as a highly disliked node may also be well connected. To take care of this situation, authors [49] correlated two attributes for each node: "Bias" and "Prestige".

5.5.1.1. Definition of Bias and Prestige

The bias of a node is the propensity to trust/ mistrust other nodes. The prestige of a node is the ultimate rank (importance) of it in compared to other nodes. Formally, let $G = (V, E)$ be a graph, where an edge $e_{ij} \in E$ (directed from node i to node j) has weight $w_{ij} \in [-1, 1]$. $d^o(i)$ and $d^i(i)$ correspondingly denote the set of outgoing links from and incoming links to node i. Bias reflects the expected weight of an outgoing edge. Using bias, the inclination of a node towards trusting/ mistrusting is measured. The bias of node i can be determined by

$$bias(i) = \frac{1}{2|d^o(i)|} \sum_{j \in d^o(i)} (w_{ij} - rank(j)) \tag{12}$$

Prestige (rank) reflects the expected weight of an in-link from an un-biased node. Intuitively, when a highly biased node (either positive or negative) gives a rating, such score should be given less importance. When a node has a positive (negative) bias and has an edge with a negative (positive) weight, that opinion should weigh significantly. Hence, the prestige (rank) of node j could be determined as -

$$rank(j) = \frac{1}{|d^i(j)|} \sum_{k \in d^i(j)} (w_{kj}(1 - X_{kj})) \tag{13}$$

where the auxiliary variable X_{kj} determines the change on weight w_{kj} based on the bias of node k.

$$X_{kj} = \begin{cases} 0 & \textit{if} \, (bias(k) \times w_{kj}) \leq 0, \\ |bias(k)| & \textit{otherwise}. \end{cases} \tag{14}$$

After each iteration of 12 and 13, edge-weight w_{kj} is scaled from the old weight as follows:

$$w_{kj}^{new} = w_{kj}^{old}(1 - X_{kj}) \tag{15}$$

5.6. Text as Graph

In order to apply the graph based ranking algorithms, we convert the text document into a graph. We extract words (except stop-words) from each sentence and represent them as nodes of our graph. Each pair of related words (lexically or semantically) forms the edges. We use SentiWordNet (a publicly available lexical resource for opinion mining) to determine the sentiment polarity of each node (signature word). SentiWordNet [8] assigns to each synset of WordNet [48] three sentiment scores: positivity, negativity, and objectivity. We choose the highest (most common) sentiment polarity of a word as the bias. Edge weights are determined by the total outgoing edges from the node. If there is a {not, no, though, but,…} present between $word_a$ and $word_b$, the edge weight (w_{ab}) receives the opposite sign of $bias_a$. Our algorithm performs the following steps:-

- Phase I: Building the Text-Graph
  - Collect signature words; use them as nodes for the graph. Use their sentiment polarity as bias.
  - Add edges between nodes (words) that reside in the same sentence (within a chosen window size).
  - Assign edge-weights (w_{ab}) based on the total outgoing edges from each source node ($word_a$).
  - Update/ add edge-weights (w_{ab}) if they are semantically related (e.g. use a matching function on their definition/synonym list).
  - Assign a random value as rank to all the nodes of the graph (initially all nodes are on the same level).
- Phase II: Applying the Model
  - Apply formula [12,13,14,15] over the graph until the $rank$ value converges.
- Phase III: Word Vectors, & Sentences
  - Create a positive word vector, W^{pos} of keywords by selecting all positively ranked words.
  - Create a negative word vector, W^{neg} of keywords by selecting all negatively ranked

words.

  - Use W^{pos} and W^{neg} to determine the weight and orientation of the sentences.
  - Group top k (can be determined by the user) negatively (positively) oriented sentences as anti-summary (summary).

The following subsections will discuss our process in detail. To demonstrate several intermediate outcomes of our process, we will use a sample article:
`https://github.com/FahmidaHamid/anti_summary_samples/blob/master/Kennedy1961/kennedyPart1.txt`,
which is a small fragment (only 77 sentences are considered) of President Kennedy's speech in 1961.

5.6.1. Signature Words

Using a standard parts of speech tagger, we extract words that are labeled as either one from the set: {noun, verb, adjective, adverb}. These are our signature words. We also collect their definition and sentiment polarity for the next phase. Table 5.1, 5.2 show the intermediate data generated from example 01.

Example 01: "The first and basic task confronting this nation this year was to turn recession into recovery."

5.6.2. Define Nodes and Edges: from a Single Sentence

Let, x and y are two words residing in the same sentence, and $|position_x - position_y| < window\ size$. We create distinct nodes (if not already exist) for x and y, and define their relations (edges) by either of the rules:

- If $parts\_of\_speech(x) = \{verb\}$, add $edge(x, y)$.
- If $parts\_of\_speech(x) \cup parts\_of\_speech(y)$
  $\subset \{noun, adjective, adverb\}$, then add $edge(x, y)$ and $edge(y, x)$.
- Finally, we add edge-weight, $w_{xy} = sign(bias(x)) \times \frac{1}{|E|}$ to all the existing edges.

5.6.3. Connect Sentences through Words: Add more Edges/ Update Weights

Let x and y are two different words from two different sentences (or from the same sentence, $|position_x - position_y| \geq window\ size$). We use their definition (available in WordNet) to determine $similarity$ between them. If $def(x)$ stands for $definition$ of x,

$$similarity(x, y) = \frac{def(x) \cap def(y)}{def(x) \cup def(y)} \tag{16}$$

We add/ update edge-weight w_{xy} and w_{yx} using the following manners:

- We do not update the graph if the $similarity(x, y)$ is $zero$.
- For an existing edge between x and y, we adjust w_{xy} as $w_{xy} + similarity(x, y) \times sign(bias(x))$.
- For a no edge between x and y, we add two new edges ($edge(x, y)$ and $edge(y, x)$) where $similarity(x, y)$ acts as the weight for the new edges.

This phase helps relate semantically closer words in the document.

To demonstrate how the graph is formed, we randomly picked two sentences from the stated article: ‘Our security and progress cannot be cheaply purchased; and their price must be found in what we all forego as well as what we all must pay’ and ‘The first and basic task confronting this nation this year was to turn recession into recovery’. The sentence graph with only these two sentences (with $window\ size = 4$) is shown in figure 5.2. We notice that word pairs {(security, recession), (security, recovery), (progress, recovery)}, for example, are connected to each other through the similarity relationship.

5.6.4. Keyword Extraction

Once the graph is built, we add a real value (can be chosen randomly) to every node as it’s $rank$. This way, there is no discrimination beforehand. Then we apply set of equations [12,13,14,15] several times (until the rank value converges) over the graph. For real time output, one can control the repetition using a threshold. Table 5.3 shows a handful of positively ranked and negatively ranked keywords (out of 568 total words) from the same article.

Table 5.1. Words & their entities

Word	PoS	Polarity	Definition
first	adj	0.0	preceding all others in time or space or degree
confront	v	-0.5	oppose, as in hostility or a competition
nation	n	0.0	a politically organized body of people under a single government
year	n	0.0	a period of time containing 365 (or 366) days
turn	v	0.0	change orientation or direction, also in the abstract sense
recession	n	0.0	the state of the economy declines; a widespread decline in the GDP and employment and trade lasting from six months to a year
recovery	n	0.0	return to an original state

Table 5.2. Degree of "Similarity"

Word	Definition	Similarity
recession	the state of the economy declines; a widespread decline in the GDP and employment and trade lasting from six months to a year	
recovery	return to an original state	0.035714
security	the state of being free from danger or injury	
progress	gradual improvement or growth or development	0.0
recovery	return to an original state	
security	the state of being free from danger or injury	0.071428

5.6.5. Sentence Extraction

Our top (positive, and negative) ranked keywords define the weights of the sentences. Let W^{pos} (W^{neg}) be the list of words achieving positive (negative) rank values. Let W^{pos} is a list of size m and W^{neg} is a list of size n.Weight of a sentence, s_j is:

$$weight(s_j^{neg}) = (\sum_{v_i \in W^{neg} \wedge v_i \in s_j} |rank(v_i)|)/(n \times |s_j|), \tag{17a}$$

$$weight(s_j^{pos}) = (\sum_{v_i \in W^{pos} \wedge v_i \in s_j} rank(v_i))/(m \times |s_j|), \tag{17b}$$

Now each sentence has two weights associated with it; $weight(s_j^{pos})$ corresponds its weight on positively ranked words whereas $weight(s_j^{neg})$ corresponds its weight on negatively ranked words. Thus S^{neg} (S^{pos}) forms a weight vector of sentences on negatively (positively) ranked ones. One

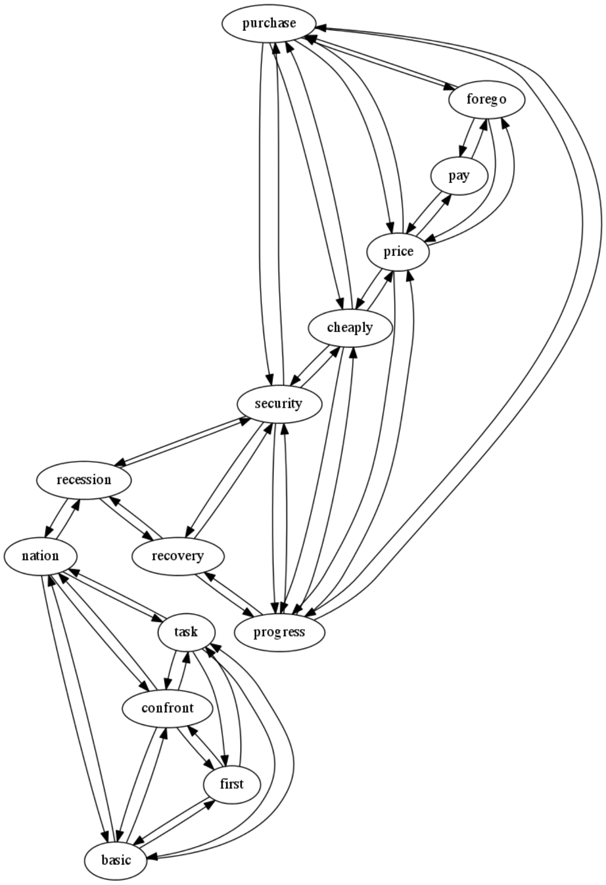


Figure 5.2. A “Sentence Graph”

Table 5.3. A subset of keywords

keyword	Rank	
initiate	1.50503400134e-07	
wisely	2.37049201066e-19	
cheaply	2.03124423939e-19	
thailand	1.52162637924e-28	
believe	2.53398571394e-38	positively
crucial	1.14205226912e-40	ranked
forego	7.26752110553e-46	
mind	3.3034186848e-46	
handicap	1.89292202562e-57	
progress	1.18505270306e-62	
cambodia	-0.0210985300569	
recovery	-0.126687287356	
recession	-0.0376108282812	
unwilling	-7.70602663247e-05	
congress	-0.064049452945	negatively
cooperate	-0.285579113282	ranked
building	-0.0387114701238	
havana	-0.0128506602917	
frontier	-0.0182778316133	
compete	-0.075853425471	

can select top k many sentences based on S^{neg} as the anti-summary. The similar line of thinking goes for generating regular summaries. To avoid promoting longer sentences, we are using length of the sentence as the normalization factor.

Table 5.4 shows two top sentences(the first one is the 2^{nd} top positively ranked and the

Table 5.4. A sample of top sentences

sentence	weight
Our security and progress cannot be cheaply purchased; and their price must be found in what we all forego as well as what we all must pay.	$1.98797587956e - 14$
The first and basic task confronting this nation this year was to turn recession into recovery.	-0.00117486599011

second one is 5^{th} top negatively ranked). The original file can be found at: `https://github.com/FahmidaHamid/anti_summary_samples/blob/master/Kennedy1961/kennedyPart1SA.txt`.

Our model uses a mutually recursive relation on bias and rank calculation. It incrementally updates the edge-weights as well. Hence, it helps get the final ranks (polarity and weights) of words on global context. Since a TextRank model does not rely on the local context of a text unit, and requires no training corpus, it is easily adaptable to other languages or domains.

5.7. Evaluation

We used TextRank (extracted) and Human (abstract) summaries from DUC 2004 (task 1) as the reference summaries. TextRank is unsupervised and it does not handle sentiment polarity. Hence, we started with hypothesis 1.

HYPOTHESIS 1. polarity based summaries and anti-summaries should almost equally intersect with TextRank generated ones.

In order to verify the hypothesis, we calculated average number of sentence intersection between each pair of the three summaries (our model generated anti-summary(N), summary(P) and TextRank summary(T)). Then we plotted them against the probability of intersection of two random generated summary. Table 5.5 explains the operations. The test cases are named as -

- case a: an average size of $(P \cap T)$,
- case b: an average size of $(N \cap T)$,
- case c: an average size of $(P \cap N)$,
- case d: an average size of $(S_1 \cap S_2)$, with any two randomly selected set S_1 and S_2 of the same size as P, N and T.

Summaries of this set of articles are stored in link:

`https://github.com/FahmidaHamid/cicling2015_dataset`

Table 5.5. Summary & their average size of intersection

Test Set	Total Files	(n)	$avg(n)$	(k sentences)	$avg(k)$	case a	case b	case c
1	163	$n > 30$	48	10	10.00	1.75	3.18	1.43
2	337	$n \leq 30$	16.5	$n/3$	5.10	1.63	1.88	1.08
				$n/2$	8.05	4.21	4.73	3.386
3	410	$n \leq 40$	20.02	$n/3$	6.34	2.626	2.304	1.44
				$n/2$	9.775	5.826	5.613	4.256

Quite interestingly, for shorter articles, case a and case b showed similar (and better) performance than case c and case d [Table 5.5, & figure 5.3]. It also supports hypothesis 1. For larger articles, case b was the winner. Section 2.3.1 gives the mathematical background for case d.

5.7.1. Baseline: Intersection of two models vs. the Random possibility

The average size of an intersection(avg) of subsets with k elements taken from a set with n elements can be determined by equation 18.

$$avg(n,k) = \frac{\sum_{i=0}^{k} i\binom{k}{i}\binom{n-k}{k-i}}{\sum_{i=0}^{k}\binom{k}{i}\binom{n-k}{k-i}} \qquad (18)$$

For two summaries of different sizes k and l this generalizes to:

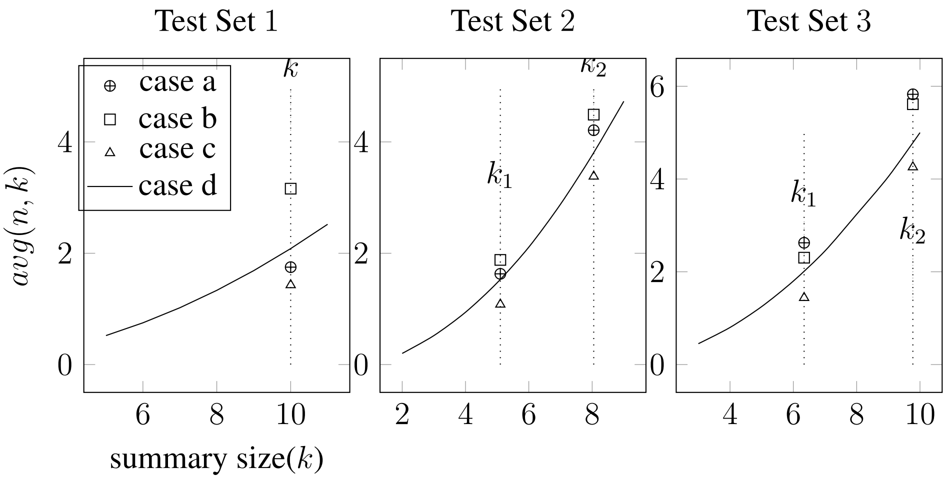


Figure 5.3. Average of sentence intersection based on Equation 4

$$avg(n,k,l) = \frac{\sum_{i=0}^{k} i\binom{k}{i}\binom{n-k}{l-i}}{\sum_{i=0}^{k}\binom{k}{i}\binom{n-k}{l-i}} \tag{19}$$

These formulas are justified as follows: Fix one of the subsets as $X = 0, 1, \ldots, k-1$. Then an intersection of size i is computed by taking a subset of X of size i (there are ($b = \binom{k}{i}$)) such sets). We have $j = l - i$ elements in X that will be selected among the b' subsets of size j of the remaining $n - k$ elements in the complement of X counted as $b' = \binom{n-k}{l-i}$. So the numerator of the fraction, will be obtained by summing up for $i = 0$ to $k - 1$ the product of i with the number of subsets b and and the number of subsets b', counting the total length of the subsets. The denominator of the fraction will count the total number of these subsets and the result of their division will give the average size of the intersections. The formula used here is another way of generating the average intersection size of two randomly generated subsets (discussed in Section 2.3.1).

Knowing the average sizes of the overlap of two summaries of size k or sizes k and l when they are different (seen as sets of words), tells us whether our model-generated summaries, and anti-summaries have a better rate of intersecting with each other (and TextRank) than random summaries would.

5.7.2. Does $(P \cap N)$ indicate something interesting?

In each case, $(P \cap N)$ is minimal (fig. 5.3) which indicates that our model is successfully splitting the two flow of sentiments from documents. This raises a set of questions, e.g.

- when $(N \cap T) \gg (P \cap T)$, should we label the article as a $negatively$ biased one?
- when $(P \cap T) \gg (N \cap T)$, should we label the article as a $positively$ biased one?
- when $(P \cap N) \gg (P \cap T)$ and $(P \cap N) \gg (N \cap T)$, is it a news/article stated from a $neutral$ point of view?

We tried to answer these questions based on experimental results. We might need voluntary human judges to label the articles based on the the extractive summaries and compare our summary based guesses. We leave this phase as a future direction. Interested reader can get our result from the following link:

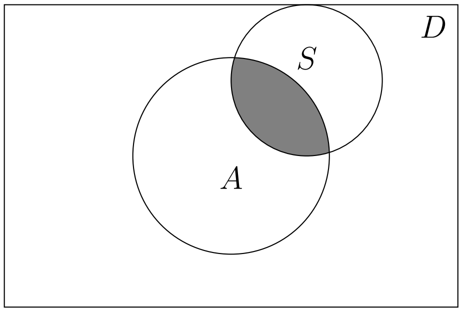
`https://github.com/FahmidaHamid/cicling2015_dataset/blob/master/summaryHalf/fileType.txt`

5.7.3. How much relevant information is retrieved?

We needed to know whether our model is gathering some relevant sentences or not. We use abstractive summaries (provided with DUC2004 dataset) and TextRank extracted ones as base results; then use the recall measure to estimate the ratio of number of relevant information retrieved.

5.7.3.1. Choosing Recall over Precision

Recall (r) is the ratio of number of relevant information received to the total number of relevant information in the system.



$$r = \frac{|A \cap S|}{|S|} \tag{20}$$

$$p = \frac{|A \cap S|}{|A|} \tag{21}$$

Another well-known measure is Precision (p) which is the ratio of number of relevant records retrieved to the total number (relevant and irrelevant) of records retrieved. If D is the original document, A is the anti-summary, and S is the standard summary, then the recall(r) and precision(p) value can be calculated from equation 20. As our test dataset had different file sizes, we can tune the anti-summary length as we want, and we cannot firmly state that $A \setminus (A \cap S)$ is irrelevant; we believe, interpretation of r is more relevant than p, in our case.

Table 5.6. Avg(recall) of P, N, T with respect to Human Summary

Test Set	Total Files	n	k	$r(P)$	$r(N)$	$r(T)$
1	163	$n > 30$	10	.469	.397	.447
2	337	$n \leq 30$	$n/3$	.408	.458	.457
			$n/2$	.545	.596	.583
3	410	$n \leq 40$	$n/3$	.449	.436	.531
			$n/2$	.588	.582	.607

Table 5.6 shows the average recall value on the our model generated summary(P), anti-summary(N), and TextRank(T) summary with respect to human (H) summary. Here n is the total number of sentences in the document, k represents the total number of sentences present in the summary. We have used unigram word matching for computing recall rate. This result gives us an idea that -

HYPOTHESIS 2. Anti-summary can help extending TextRank summary in order to produce sentiment oriented summary.

5.7.4. Evaluation through examples: ($(P \cap N) \approx \emptyset$)

The next block shows a Summary and an Anti-summary produced by our system, for the data file:

`https://github.com/FahmidaHamid/anti_summary_samples/blob/master/googleCase.txt`.

This example shows a clear distinction between the summary and the anti-summary sentences. The summary sentences represent the view of European Union and other Companie's questioning Google's privacy policy. On the other hand, the anti-summary sentences are about Google's steps and clarifications in the issue. So, anti-summary is a better approach to generate upstream information from a document, without knowing the topic(s) in the document.

summary:
"While there are two or three open minds on the company's advisory group that oversees the exercise, the process appears to be fundamentally skewed against privacy and in favor of publication rights". Its advisers include Wikipedia founder Jimmy Wales who has described the right as "deeply immoral," according to a report in the Daily Telegraph, as well as a former Spanish privacy regulator and an ex-justice minister of Germany. "It doesn't help to throw around big, loaded words like that when you're trying to find a convergence of views". "I hope Google take the opportunity to use these meetings to explain its procedures and be more transparent and receptive about how it could meet the requirements of this judgment," said Chris Pounder, director of Amberhawk, a company that trains data-protection officials. Anyone interested in attending can sign up online about two weeks before the events, Google said.

anti_summary:
Google chairman Eric Schmidt and Drummond are among the advisers who will draft a report on the ruling to discuss the implications of the court case for Internet users and news publishers and make recommendations for how the company should deal with requests to delete criminal convictions. Privacy regulators have criticized Mountain View, California-based Google's steps to tell Web publishers when it is removing links to their sites. Regulators are drafting guidelines on how they should handle any disputes by people who were unhappy at how Google handles their initial request for links to be removed. The first event takes place in Spain, the trigger for the EU court ruling that changed Google's business when the company fought an order by the country's data-protection regulator to remove a link with details to a state auction of houses to cover tax debt that popped up on a search for Mario Costeja Gonzalez. Al Verney, a spokesman for Google in Brussels, said the company will hear from invited experts and also from people in the audience at the events, who could sign up on the Internet to attend.

5.7.5. Evaluation through examples: ($P \approx N$)

We would like to show another sample Summary and Anti-summary which are generated from a news over the aid provided to the flood-damaged area in Honduras and Nikaragua. The news can be found at:

`https://github.com/FahmidaHamid/anti_summary_samples/blob/master/testData02/APW19981106.0869`.

It shows two important features:

- One, out of pulled top three sentences is common between summary and the anti-summary.
- The summary sentences are mostly about aid, whereas anti-summary sentences are about damage and delaying on delivering the foods to the sufferers.
- Hence it is a good example of non-polar articles.

summary:
In the Aguan River Valley in northern Honduras, floodwaters have receded, leaving a carpet of mud over hundreds of acres (hectares). A score of cargo aircraft landed Thursday at the normally quiet Toncontin airport in the Honduran capital of Tegucigalpa, delivering aid from Mexico, the United States, Japan and Argentina. First lady Hillary Rodham Clinton added Nicaragua and Honduras to a trip she plans to the region beginning Nov. 16.

anti_summary:
Foreign aid and pledges of assistance poured into Central America, but damage to roads and bridges reduced the amount of supplies reaching hundreds of isolated communities to a trickle: only as much as could be dropped from a helicopter, when the aircraft can get through. A score of cargo aircraft landed Thursday at the normally quiet Toncontin airport in the Honduran capital of Tegucigalpa, delivering aid from Mexico, the United States, Japan and Argentina. "It's a coincidence that the ships are there but they've got men and equipment that can be put to work in an organized way," said International Development Secretary Clare Short.

From these two examples, we can state that:

HYPOTHESIS 3. Summary and Anti-summary overlap at a significant amount, if the article contains more objective sentences than subjective ones.

Besides SentiWordNet, we search for more accurate sentence and word-level sentiment analyzer. Mao and Lebanon[40]'s work focuses on a supervised model of sentence level sentiment detection. We can adopt their technique, apply sentence level sentiment as the bias and then rank the sentences. One important aspect of working with text is noise reduction. Not handling anaphora resolution is the weakest point for our experiments. But one can easily modify our graph generation approach to get rid of this issue.

5.8. Future Directions

In future, we would like to apply this graph based technique as a semi-supervised approach. Using some sentiment training dataset, we can adjust the bias of each node in the graph, and then use a sentiment classifier or SentiWordNet to define the direction of polarity. Besides, we will be applying anaphora resolution techniques and semantic parsing while defining the graph. For shorter articles, we have applied anaphora resolution by hand. It performed better on defining sentence connectivity more accurately and ranked related words more precisely. We also plan to extend this work and build a model that can generate summary not only by extracting sentences but also by rephrasing some of them.

5.9. Summary of the Chapter

Our approach is domain independent and unsupervised. Using this graph-based model we mainly generate a set of extrinsic summaries that covers facts with sentiments. Then using the baseline (Hypothesis 1, Chapter 2) we devise an evaluation mechanism of the informativeness of the summaries. The baseline gives us an idea of minimum expected overlap between the sentiment-biased and non-sentiment-biased summaries. Our experiments cover several interesting scenarios; but the most import finding of the chapter is: we apply the baseline to detect the quality of an extrinsic summarization task when we do not have references generated by human evaluators.

In Chapter 6, we deviate our attention from the evaluation methodology and discuss a framework as part of our future research. The framework adopts the graph-based technique that we have used in this chapter as it has been successful to tell apart the two poles of sentiments from the document.

CHAPTER 6

GENERATING TOPIC-SPECIFIC SENTIMENT LEXICONS: A FRAMEWORK

6.1. Introduction

A typical approach to sentiment analysis is to start with a lexicon of positive and negative words and phrases. Several lexicons are publicly available for researchers: Bing Liu's opinion lexicon, MPQA subjectivity lexicon, SentiWordNet, Harvard General Inquirer etc. In those lexicons, entries are tagged with a prior polarity: out of context, does the word seem to evoke something positive or something negative [68]. Rao and Ravichandran [62] worked on three different languages: English, French, Hindi to show the polarity detection as a semi-supervised label propagation problem in a graph. But they did not consider context or topic while building the lexicon.

The contextual polarity of the phrase where a word appears may be different from the prior polarity. Hence, Wilson et al. [68] introduced some features, by using which they were able to detect contextual polarity of phrases with 65.7% accuracy. Later the authors explored that there is a difference between prior and contextual polarity [69]: words that lose their polarity because of the context, or whose polarity is reserved because of the context.

One of the simple yet well-known domain specific sentiment lexicon generation approaches was done by [56]. They started with a seed sentiment lexicon. From a collection of documents in a domain, they gathered features using the seed-words as they followed the assumption that sentiment words are almost always associated with features; and then gathered more seeds from the features and so on, until the system could not find any other new seed.

We present, in this work, a graph based technique to generate sentiment lexicon which starts with a seed set of sentiment words and probable context-word set for a single topic. The edge and node weights of the graph are determined based on the sentence structure, semantic relatedness of phrases, user provided numeric tag (we call them 'star-rating') per review and the seed lexicon's prior polarity. Another distinguishable feature of our model is, graph nodes are not single sentiment phrases, they are pairs or tuples made up of sentiment-words and context-words. Finally, we plan to publish a set of tuples (e.g. (context-word, sentiment-word)) along with intensity of polarity as the topic-specific context-aware sentiment lexicon.

We will be adapting graph based ranking method [49] to evaluate the polarity of the pairs. When the polarity (expressed as ranks) of the tuples converges, we present the tuples as a new set of sentiment lexicon. Then we can apply the lexicon information to predict the star-ratings of test reviews and compare our result with the user given stars. Instead of trying out bootstrapping method as Qiu Guang [56] did, we believe our approach should perform better on defining the intensity of the polarity as graph-based ranking algorithm is a way of deciding on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information [45].

6.2. Related Work

Kaji et al. [23] described a technique on building lexicon from a massive collection of html documents for Japanese language. Their key idea was to develop structural clues in order to achieve high precision at the cost of low recall. They have used some “cue” words, topic markers, etc. to detect polar sentences. According to two human judges, the precision of their system was 91.4% and 92%. During the error analysis phase, authors stated that most of the errors were caused by lack of context. Table 6.1 shows how the polarity of sentiments changes with the context and topic. Yue Lu [34] focused on the problem of building a sentiment lexicon which is domain specific and dependent on the aspect in context, given an unlabeled opinionated text collection. Valentin

Table 6.1. Orientation of “Sentiment Polarity” due to “Context” and “Topic”

Sentiment Word	Sentence	Prior Polarity
large	The pc has a $large^{+}$ monitor	+ve
	The chip is $large^{-}$ at size	-ve
unpredictable	Cricket is an $unpredictable^{+}$ game	+ve
	The weather of Texas is really $unpredictable^{0}$	neutral

at al. [19] describe a bootstrapping method for generating topic-specific lexicon from a general purpose polarity lexicon.

6.3. Our Approach

We focus on producing topic-specific sentiment lexicon considering the context words. There are several online resources that have product reviews (amazon.com, tripadvisor.com, the IMDB* movie review collection, etc). Given a specific topic, we collect a moderately large number of reviews on the topic from these sites.

6.3.1. Grouping Reviews in Clusters

We cluster reviews with the same label of stars $\{r_0, r_1, \dots\}$ into a single group. This helps us determine the $weight$ of the sentiment words. If the system accepts star-rating from 1 to x,(assume, increasing order indicates higher positive polarity); we divide the scale into three sections:

- Reviews with Negative Sentiment: if the star rating is $\leq x/3$
- Reviews with Positive Sentiment: if the star rating is $\geq 2x/3$
- Neutral Reviews: $x/3 <$ star-rating $< 2x/3$

We keep neutral reviews as well as the positive and negative ones since they provide support to the polar ones.

6.3.2. Defining the Contexts

The next challenge is to find out the contexts related to the topic. Unlike Yue [34], we do not manually label the aspects for the topic; otherwise it would be infeasible to build a system with very large scale dataset. We plan to detect keyphrases from the document sets we used in Section 6.3.1. Several approaches (Unsupervised: TextRank [45], LDA [3] or Supervised: MAUI [44]) can be used for the keyword detection phase. The keywords will act as the context of the specific topic.

*`http://www.imdb.com/`

6.3.3. Preparing “(context-word, sentiment-word)” tuple

From the detected keyphrases and a seed set of sentiment-lexicon, we build a set (V) of ordered pairs (a, b), where $a \in$ keyphrases and $b \in$ sentiment-lexicon. We build (a, b) as a tuple if they appear in the same sentence (or within a clause). We label the polarity of the tuple by considering the prior polarity in the seed lexicon and some simple intuitive rules (e.g. Table 6.2).

Table 6.2. Orientation of tuple (a,b); 0 indicates neutral sentiment

Example	a	b	(a, b)
The PC has a $large_b$ $monitor_a$	0	+	+
I $don't\ like_b$ the $design_a$ of the purse.	0	-	-
I am $very\ optimistic_b$ about the $qualities_a$ of product A.	+	+	+
The $qualities_a$ of product A are $horrible_b$.	+	-	-
The $blurry\ screen_a$ of the phone $disappionted_b$ me.	-	-	-

6.3.4. Connecting Tuples

Now we need to create the connection (i.e., edge) between the tuples. We follow the rules to create connections:

- tuple (a_1, b_x) connects to(and from) tuple (a_1, b_y), given $b_x \neq b_y$.
  - if, in the original document (i.e. review), (a_1, b_x) and (a_1, b_y) are joined using {and, as well, if . . . then}, then they will have positive weighted links.
  - if (a_1, b_x) and (a_1, b_y) are joined using { but, though, although, yet, etc.}, then they will have negative weighted links.
- If a_i and a_j has some hypernym or synonym relationship, then tuple (a_i, b_x) and (a_j, b_y) will be connected through positive links.
- If a_i and a_j form antonym relationship, then tuple (a_i, b_x) and (a_j, b_y) will be connected through negative links.
- tuple (a_i, b_x) and tuple (a_j, b_x) will be connected through positive links.

Now we have a weighted graph $G = (V, E)$ with nodes (i.e. tuples) and edges (connections between nodes). Based on the progress of our project, we will add, eliminate, or modify the techniques stated here.

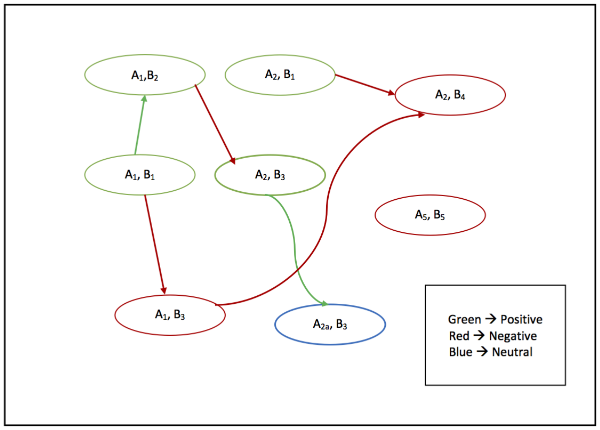


Figure 6.1. A polar-graph built with multiple reviews on a single topic

6.4. The Polarity based Rank Model

We initially planned to work with two graph based models: TextRank and Trust-based Network [49]. Eventually we explored several cases that TextRank cannot handle (e.g. negative or zero weighted edges, more than one property for a node etc.) Hence, we will propose our model by adapting the trust-based network.

6.4.1. Trust-based Network: A brief Introduction

We have discussed the model in Section 5.5.1. We will be using the same model to represent this polar graph. Each tuple (a, b) will act as the node of the graph. And the co-occurrence of the tuples in the same document (or review) will connect them. We will be considering the notion of popular conjunctions, e.g, and, or, but, though, etc. to detect the type (positivity or negativity) of the edges. The $bias$ of each tuple (a, b) will be determined by the "star-rating" associated with it and the direction of polarity is determined by rules similar to table 6.2. If same tuple is found at different star-rated classes, we will preserve the most frequent one. Each tuple (node) will initially be assigned with same $rank$ value.

6.4.2. Calculating the Rank

In order to update the ranks we call formula 13. The next phase is to update the bias; and to do so, we use equation 12. In between, equation 14 and equation 15 are applied to adjust the edge weights. The order of updating bias and rank is not relevant. We can start with either one of them; and recursively call one after another until the $rank(a, b)$ for all pairs (a, b) converges. Then we publish the pairs along with their ranks as the context based topic focused sentiment lexicon.

6.5. Evaluation

We believe, it will be a good start, if we can get a fairly large dataset; and given one topic, we randomly select 90% reviews on the topic to develop our sentiment-lexicon and apply our lexicon to the rest 10% for predicting the star-ratings. Based on our prediction and the user given rating, we should be able to see the performance of our system. We will be using dataset from [43] since they already have a large collection (a total of 143,663,229) of user rated review sets on different categories (books, computer, kitchen appliances, clothes, etc.)

6.5.1. Predicting Star-Ratings

A handful of qualitative research has been done on 'predicting' the star rating from the reviews. Authors [12] handled this task as both a regression and a classification modeling problem and explored several combinations of syntactic and semantic features. They suggested that classification techniques perform better than ranking modeling when handling evaluative text. At this preliminary stage, we plan to use some standard machine learning techniques (Naive-Bayes, Decision Tree, SVM) with the training dataset, with our system generated lexicons in order to predict the star-rating for the test reviews.

Another interesting research [9] talks about text-derived information in predicting the rating of a review in a recommendation system. The authors primarily implemented their system for hotel reviews. We would like to use their method as a state-of-the art; compare our polarity-biased model provided ratings with their system generated ones. That way, we will have two references (human-provided, system-generated ones) and our model generated ratings to test the rating distance between each pair. This can give us a scope or idea to improve the technique of generating topic-focused words.

6.5.2. Distance Function: How close can we predict?

Given a gold standard, precision, recall, and f-measure are three mostly used techniques for comparing the performance. To determine the quality of our system they are not the best fit. For example, if the system predicts 3 star whereas the user provided 4, we should not label it as an entirely wrong prediction. We would like to measure the distance between our rating and the user provided one.

Based on the topic, distance function can change its characteristics. It can be linear, exponential, or logarithmic. Here we show one linear distance function that we will use for our system.

6.5.3. A Linear Distance Function

Let U be the set of user ratings, S be the set of system generated ratings, then the average distance function $avg(D)$ tells us to what extent our model is deviating from the gold standard.

$$U = \Big(u_1, u_2, \ldots, u_n\Big)$$

$$S = \Big(s_1, s_2, \ldots, s_n\Big)$$

$$D = \Big(d_1, d_2, \ldots, d_n\Big)$$

where

$$d_i = abs(u_i - s_i)$$

$$avg(D) = \frac{1}{n}\sum_{i=1}^{n} d_i$$

We can set some threshold (t), and select the reviews for who the $abs(u_i - s_i) \geq t$. Then we feed them in the training set, update our lexicon and generate new S to see whether the performance improves or not. This way we can apply some cross-validation techniques to improve the system performance.

6.6. Summary of the Chapter

This framework is been designed based on the related works already done for context-oriented sentiment lexicon generation. The problem first caught our notice while we explored Bin Liu's sentiment lexicon which is just two large list of positive and negative words; and the SentiWordNet, which has some polarity value associated with each sentiment-word according to the parts of speech tags. But the polarity of a sentiment-word depends on the topic (domain, at large scale) and the context (the words surrounding the sentiment words). In most cases, we can easily identify the sentiment-words (which are mostly adjectives and adverbs), but their polarity (positive, negative, or neutral) and intensity of polarity varies based on the context.

Generalizing sentiment for all domains is not intuitive as well. Hence we aim at designing an automated system that can build some domain (or topic, to be more precise) specific sentiment lexicon which extracts sentiment oriented statements from a large group of unknown people (hence it is biased towards any specific race) and employs a mutually recursive technique to converge towards the intensity and direction of polarity.

In Chapter 7, we draw the concluding remarks of our achievements and also outline some future research directions.

CHAPTER 7

CONCLUSION

Text summarization and keyphrase extraction are challenging tasks. Maintaining linguistic quality, optimizing both compression and retention, all while avoiding redundancy and preserving the substance of a text, is a difficult process. Equally difficult is the task of evaluating such outputs. Interestingly, a summary generated from the same document can be different when written by different humans (or by the same human at different times). Hence there is no convenient and complete set of rules to test machine-generated outputs. In this Chapter, we discuss the major issues that have been addressed through our work followed by our proposed solutions. We also refer to some of the areas which need more research attention in order to handle complicated cases.

7.1. Research Problem and Proposed Solutions

We started with the aim of grafting different environmental aspects to the evaluation tool for automatic summarization and keyphrase extraction tasks. We came across several issues. The stated ones are some of the significant affairs resolved by our proposed methodology:

7.1.1 The Generic Baseline

There is no standard rule to define the baseline for an evaluation process. For summarization, a couple of sentences from the first paragraph (news articles) or from the last paragraph (scientific articles) are usually considered as the baseline. The situation is fuzzier with evaluating keyphrase extraction task. So the baseline is dependent on the structure of the document and varies for the same algorithm tested over different datasets. We argue that the baseline should be related to the original document, the gold standard, and the produced output. In our approach, a baseline is considered as the ratio of the multiplication of the total number of terms present in the system-generated output and the reference output to the total number terms present in the original document (equation 2) . We have proven that, the ratio is the average size of intersection of two randomly generated outputs with the size of the system-generated summaries and the human-written summaries. This hypothesis helps us getting rid of pre-defined length constraint.

7.1.2 The Relativized Scale

Since we allow the systems and humans to produce outputs with different lengths, we need a way to relativize it so that the scores of different systems become comparable. In our evaluation approach, the overlap between two summaries is compared against the average intersection size of the randomly generated baselines. This relativized scale is named $i\text{-}measure$ (equation 3). In Section 2.4.1, we have shown a relation between the absolute scale ($f\text{-}measure$) and the proposed relativized scale ($i\text{-}measure$).

7.1.3 Handling the Subjectivity of Multiple Evaluators

The evaluation paradigm falls short when human written references are not available and the paradigm becomes less accurate when only a single model is available [33]. Recent conferences (TAC, SemEval, etc.) have exploited more time and resources to gather multiple and qualitative reference-outputs. In some cases, human evaluators are given proper training before they perform the tasks. Therefore the manual outputs are either considered equally important (ROUGE), or they are weighted based on the similarity (Pyramid) with each other. Interestingly, when a single human was assigned to score the human annotated outputs (e.g. DUC 2004, Task 05), in most of the cases, he could not provide the same score to all the human participants for the same task. Considering this, we would like to state that both the disagreements and the agreements are equally important, and we need a robust technique to adjust to their variations.

$i\text{-}measure$ is a resilient approach that reduces the subjectivity of the evaluators. We propose the ranking mechanism of machine-generated summaries based on the concept of closeness with respect to multiple reference summaries. The key idea of our methodology is the use of normalized weighted relatedness of each reference to the other. The aggregated normalized relatedness of one reference produces its confidence score. The system output is compared against every reference separately, multiplied by the corresponding confidence score. Finally, the partial scores are added up to get the total score (equation 9). When the test set contains multiple documents, we average the total score to produce the final rank. ROUGE, the most popular approach so far, has been criticized as the numeric values given to the systems by ROUGE are merely distinguishable from one another. In the Pyramid approach, summaries with the less frequent terms are penalized over the summaries with more frequent terms. As we compare the system output with each reference separately and use the confidence score as a weighing factor for the reference, we consider the more-frequent and the less-frequent terms equally important for the system-generated output.

7.1.4 Introducing Semantic Relatedness

It is likely for humans to deliver abstractive outputs. Humans also use synonyms, hypernyms, hyponyms, or paraphrases to avoid mimicry as much as possible. On the contrary, most of the automated systems are either entirely extractive, or they have not been very proficient at generating new phrases like humans do. Due to these challenges, uni-gram word matching is popular with the evaluation techniques. It can be noticed that in order to use the Pyramid approach the evaluation conference had to employ manual laborers to identify similar clauses before putting them in different layers of the pyramid. ROUGE, on the other hand, uses exact string matching techniques for comparing n-grams. However, the question still remains open: if some parts of the gold standard are missing in the original document, there is completely no way for an extractive approach to find them. And, it has a direct impact on the absolute scale. We have also made efforts to produce an equivalence class for each word with the help of a standard knowledge source (e.g. WordNet). We suggest to use different forms of semantic relatedness to accommodate with abstractive outputs.

7.1.5 The Quality of Summaries and Keyphrases

The standard methods (precision, recall, f-measure) are fair if all the outputs are truncated to the same length. Thus the approaches we have experienced so far (e.g. DUC 2004, TAC 2009, SemEval 2010), focus on generating more informative, but less-fluent summaries and keyphrases to fit in the pre-defined length. Readability, one of the major goals, is thus neglected. In Section 2.4.2, we have clearly discussed how $i\text{-}measure$ adjusts to the variation of lengths between different system outputs and reference outputs. One of the major advantages of being lenient towards the length is that it will encourage the automated systems to produce grammatically well-formed sentences. Thus, our evaluation approach will increase the possibility and scope of producing fluent and more informative outputs. We will link a reliable and efficient syntax parser to the evaluation tool that is already published in the GitHub repository: `https://github.com/FahmidaHamid/code_ictir`.

7.1.6 The "Diamond Standard" Dataset

The GitHub repository* comes with a collection of scientific articles with author provided abstracts and keywords to be used as the baselines. We believe, the dataset can be used by the research community for testing the performance of their designed algorithms. We have also integrated a version of TextRank algorithm and it's corresponding output set (keywords, and summaries). TextRank is a credible state-of-the-art algorithm and it's output can be considered as a system-generated baseline for the dataset. The repository may also be used by others to employ human evaluators for producing a parallel set of standard output and thus enriching the dataset.

7.1.7 Evaluating an Extrinsic Summarization Task

Besides using the dataset and system outputs from some previous conferences, we have performed a case study. For example, we have used the DUC 2004 conference to test the performance of our proposed system. Most of the tasks for the conference can be categorized as intrinsic tasks. We wanted to test the quality of our evaluation approach for some extrinsic tasks, for example, question-specific summarization, domain-specific keyphrase extraction, etc. Besides that, we wanted to minimize the involvements of human evaluators. Therefore we have developed an algorithm to generate a sentiment-oriented summary from a document.

We have redefined a trust-based model to represent text as a connected graph, not only through the co-occurrence, or semantic-relatedness, but also using the sentiment biased-ness of some units (e.g. words, sentences). Our aim is to determine two different summaries from a single document: one representing positive sentiments, and the other representing negative sentiments. The approach is unsupervised; hence it does not require any training dataset. While prior work

*`https://github.com/FahmidaHamid/dataset_ictir`

exists on summarizing sentiment-biased articles (especially reviews of movies/books/electronic-items, etc.). Nonetheless, extracting sentiment-polar sentences from a single document by recursively considering the impact of one pole of sentiment to the other is a novel idea.

As a state-of-the art reference algorithm, we use TextRank to produce summaries without considering the sentiments. We design a test-set to evaluate the informativeness of the sentiment-polar summaries with respect to the TextRank generated summaries. From that particular set of experiments we can see that the model could extract polar summaries without overlapping excessively with each-other. The intersection between two polar summaries is always below the random generated baseline (equation 2).

7.2. Future Directions

The next challenge, we feel, is to devise an evaluation approach that can work without human references. We, at least for now, want to make the best use of one reference per document (or document cluster) to produce a margin for the divergence of the probability distribution (e.g. using Jensen Shannon divergence) between the document and the human-written standard using the diamond standard dataset. We can use this distribution as a reference for comparing the probability distribution between system-generated output and the original document.

The future plans are can be outlined as the followings: the technique has to be revised so that it correlates better with the human decisions, pre-determines some weighted relatedness to query-based or domain-specific terms, and becomes more adaptable for evaluating extrinsic tasks; and also it has to be extended for handling the ranking process without human references.

Through the case study explained in Chapter 5, we have been able to extract some qualitative and disjoint summaries with the trust-based model. Therefore we plan to reuse it in order to deliver topic-focused sentiment lexicons, considering the topic-related words given as the context. We have, so far, a few human annotated generic sentiment lexicons (e.g. SentiWordnet, Harvard General Inquirer, MPQA Subjectivity Lexicon, etc.) which are relatively small in size. Our plan is to use the dataset from [43] along with the framework (Section 5.5.1) to publish a set of sentiment lexicons per topic. We also want to publish an automated tool that can provide and update topic-focused sentiment lexicons, once fed with a moderately large dataset from different sources.

BIBLIOGRAPHY

[1] Alexandra Balahur, Mijail Alexandrov Kabadjov, Josef Steinberger, Ralf Steinberger, and Andrs Montoyo, *Summarizing opinions in blog threads.*, PACLIC (Olivia Kwong, ed.), City University of Hong Kong Press, 2009, pp. 606–613. 45

[2] Philip Beineke, Trevor Hastie, Christopher Manning, and Shivakumar Vaithyanathan, *Exploring sentiment summarization*, Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications (Yan Qu, James Shanahan, and Janyce Wiebe, eds.), AAAI Press, 2004, AAAI technical report SS-04-07. 45

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, *Latent dirichlet allocation*, J. Mach. Learn. Res. 3 (2003), 993–1022. 67

[4] Jaime Carbonell and Jade Goldstein, *The use of mmr, diversity-based reranking for reordering documents and producing summaries*, In SIGIR, 1998, pp. 335–336. 4

[5] Jean Carletta, *Assessing agreement on classification tasks: The kappa statistic*, Comput. Linguist. 22 (1996), no. 2, 249–254. 12

[6] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, *Natural language processing (almost) from scratch*, J. Mach. Learn. Res. 12 (2011), 2493–2537. 41

[7] Günes Erkan and Dragomir R. Radev, *Lexrank: Graph-based lexical centrality as salience in text summarization*, J. Artif. Int. Res. 22 (2004), no. 1, 457–479. 4, 13, 47

[8] Andrea Esuli and Fabrizio Sebastiani, *Sentiwordnet: A publicly available lexical resource for opinion mining*, In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC?06, 2006, pp. 417–422. 52

[9] Gayatree Ganu, Nomie Elhadad, and Amlie Marian, *Beyond the stars: Improving rating predictions using review text content*. 70

[10] Dan Gillick and Yang Liu, *Non-expert evaluation of summarization systems is risky*, Proceedings NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, 2010, pp. 148–151. 6

[11] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell, *Summarizing text documents: Sentence selection and evaluation metrics*, Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '99, ACM, 1999, pp. 121–128. 17, 24, 25, 39, 47

[12] Narendra Gupta, Giuseppe Di Fabbrizio, and Patrick Haffner, *Capturing the stars: Predicting ratings for service and product reviews*, Proceedings of the NAACL HLT 2010 Workshop on Semantic Search (Stroudsburg, PA, USA), SS '10, Association for Computational Linguistics, 2010, pp. 36–43. 70

[13] Udo Hahn and Inderjeet Mani, *The challenges of automatic summarization*, Computer 33 (2000), no. 11, 29–36. 2, 11

[14] Fahmida Hamid, David Haraburda, and Paul Tarau, *Evaluating text summarization systems with a fair baseline from multiple reference summaries*, Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings (Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello, eds.), Lecture Notes in Computer Science, vol. 9626, Springer, 2016, pp. 351–365. 14, 16, 17, 24, 36, 39, 42

[15] Fahmida Hamid and Paul Tarau, *Anti-summaries: Enhancing graph-based techniques for summary extraction with sentiment polarity*, Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part II (Alexander F. Gelbukh, ed.), Lecture Notes in Computer Science, vol. 9042, Springer, 2015, pp. 375–389. 9, 45

[16] Eduard Hovy and Chin-Yew Lin, *Automated text summarization and the summarist system*, Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998 (Stroudsburg, PA, USA), TIPSTER '98, Association for Computational Linguistics, 1998, pp. 197–214. 2, 3, 35, 47

[17] Eduard Hovy, Chin yew Lin, Liang Zhou, and Junichi Fukumoto, *Automated summariza-*

tion evaluation with basic elements, In Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC, 2006. 26

[18] Anette Hulth, *Improved automatic keyword extraction given more linguistic knowledge*, Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (Stroudsburg, PA, USA), EMNLP '03, Association for Computational Linguistics, 2003, pp. 216–223. 37

[19] Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp, *Generating focused topic-specific sentiment lexicons*, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (Stroudsburg, PA, USA), ACL '10, Association for Computational Linguistics, 2010, pp. 585–594. 67

[20] Hongyan Jing, Regina Barzilay, Kathleen Mckeown, and Michael Elhadad, *Summarization Evaluation Methods: Experiments and Analysis*, In AAAI Symposium on Intelligent Summarization, 1998. 11

[21] Karen Spärck Jones, *Towards better nlp system evaluation*, HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994, 1994, pp. 102–107. 11, 24

[22] Karen Sparck Jones, *Automatic summarising: factors and directions*, CoRR cmp-lg/9805011 (1998). 4

[23] Nobuhiro Kaji and Masaru Kitsuregawa, *Building lexicon for sentiment analysis from massive collection of HTML documents*, Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 1075–1083. 66

[24] M. G. Kendall, *A new measure of rank correlation*, Biometrika 30 (1938), no. 1/2, pp. 81–93 (English). 31

[25] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin, *Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles*, Proceedings of the 5th International Workshop on Semantic Evaluation (Stroudsburg, PA, USA), SemEval '10, Association for Computational Linguistics, 2010, pp. 21–26. 37

[26] Omer Levy and Yoav Goldberg, *Dependency-based word embeddings*, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Baltimore, Maryland), Association for Computational Linguistics, June 2014, pp. 302–308. 41

[27] C. Y. Lin, *Looking for a few good metrics: Automatic summarization evaluation - how many samples are enough?*, Proceedings of the NTCIR Workshop 4, 2004. 24

[28] Chin-Yew Lin, *Rouge: a package for automatic evaluation of summaries*, 2004, pp. 25–26. 14, 25, 26, 39

[29] Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie, *An information-theoretic approach to automatic evaluation of summaries*, Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (Stroudsburg, PA, USA), HLT-NAACL '06, Association for Computational Linguistics, 2006, pp. 463–470. 13

[30] Chin-Yew Lin and Eduard Hovy, *Automatic evaluation of summaries using n-gram co-occurrence statistics*, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 (Stroudsburg, PA, USA), NAACL '03, Association for Computational Linguistics, 2003, pp. 71–78. 24, 37

[31] Bing Liu, *Sentiment analysis and opinion mining*, Synthesis Lectures on Human Language Technologies 5 (2012), no. 1, 1–167. 49

[32] Edward Loper and Steven Bird, *Nltk: The natural language toolkit*, Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1 (Stroudsburg, PA, USA), ETMTNLP '02, Association for Computational Linguistics, 2002, pp. 63–70. 21

[33] Annie Louis and Ani Nenkova, *Automatically assessing machine summary content without a gold standard*, Comput. Linguist. 39 (2013), no. 2, 267–300. 13, 25, 74

[34] Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai, *Automatic construction of a context-aware sentiment lexicon: An optimization approach*, Proceedings of the 20th

International Conference on World Wide Web (New York, NY, USA), WWW '11, ACM, 2011, pp. 347–356. 66, 67

[35] H. P. Luhn, *The automatic creation of literature abstracts*, IBM J. Res. Dev. 2 (1958), no. 2, 159–165. 46

[36] Inderjeet Mani, *Advances in automatic text summarization*, MIT Press, Cambridge, MA, USA, 1999. 3

[37] Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth Sundheim, *The tipster summac text summarization evaluation*, 1999. 7

[38] Inderjeet Mani and Mark T. Maybury, *Automatic summarization*, Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Companion Volume to the Proceedings of the Conference: Proceedings of the Student Research Workshop and Tutorial Abstracts, July 9-11, 2001, Toulouse, France., 2001, p. 5. 1, 3, 24

[39] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, *Introduction to information retrieval*, Cambridge University Press, New York, NY, USA, 2008. 16

[40] Yi Mao and Guy Lebanon, *Isotonic conditional random fields and local sentiment flow*, Advances in Neural Information Processing Systems, 2007. 63

[41] Daniel Marcu, *From discourse structures to text summaries*, In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, 1997, pp. 82–88. 24

[42] Daniel Marcu, *The automatic construction of large-scale corpora for summarization research*, Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '99, ACM, 1999, pp. 137–144. 35

[43] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel, *Image-based recommendations on styles and substitutes*, Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '15, ACM, 2015, pp. 43–52. 70, 77

[44] Olena Medelyan, Eibe Frank, and Ian H. Witten, *Human-competitive tagging using automatic keyphrase extraction*, Proceedings of the 2009 Conference on Empirical Methods in

Natural Language Processing: Volume 3 - Volume 3 (Stroudsburg, PA, USA), EMNLP '09, Association for Computational Linguistics, 2009, pp. 1318–1327. 67

[45] R. Mihalcea and P. Tarau, *TextRank: Bringing order into texts*, Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing, July 2004. 4, 36, 47, 50, 66, 67

[46] Rada Mihalcea, *Graph-based ranking algorithms for sentence extraction, applied to text summarization*, In Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, ACLdemo 04, 2004. 50

[47] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, *Distributed representations of words and phrases and their compositionality*, Advances in Neural Information Processing Systems 26 (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), Curran Associates, Inc., 2013, pp. 3111–3119. 41

[48] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, *Introduction to WordNet: an on-line lexical database*, International Journal of Lexicography 3 (1990), no. 4, 235–244. 14, 41, 52

[49] Abhinav Mishra and Arnab Bhattacharya, *Finding the bias and prestige of nodes in networks based on trust scores*, Proceedings of the 20th International Conference on World Wide Web (New York, NY, USA), WWW '11, ACM, 2011, pp. 567–576. 48, 50, 66, 69

[50] Andriy Mnih and Geoffrey E. Hinton, *A scalable hierarchical distributed language model*, Advances in Neural Information Processing Systems 21 (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), Curran Associates, Inc., 2009, pp. 1081–1088. 41

[51] Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown, *The pyramid method: Incorporating human content selection variation in summarization evaluation*, ACM Trans. Speech Lang. Process. 4 (2007), no. 2. 15, 25, 39

[52] Ani Nenkova and Rebecca J. Passonneau, *Evaluating content selection in summarization: The pyramid method*, HLT-NAACL, 2004, pp. 145–152. 25, 27

[53] Ani Nenkova and Lucy Vanderwende, *The impact of frequency on summarization*, Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101 (2005). 26

[54] Thuy Dung Nguyen and Min-Yen Kan, *Key phrase extraction in scientific publications*, Proceeding of International Conference on Asian Digital Libraries, 2007, pp. 317–326. 37

[55] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, *Bleu: A method for automatic evaluation of machine translation*, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (Stroudsburg, PA, USA), ACL '02, Association for Computational Linguistics, 2002, pp. 311–318. 26

[56] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen, *Expanding domain sentiment lexicon through double propagation*, Proceedings of the 21st International Jont Conference on Artifical Intelligence (San Francisco, CA, USA), IJCAI'09, Morgan Kaufmann Publishers Inc., 2009, pp. 1199–1204. 65, 66

[57] D. R. Radev, E. Hovy, and K. McKeown, *Introduction to the special issue on summarization*, Computational Linguistics 28 (2002), no. 4, 399–408. 12

[58] D.R. Radev, S. Blair-Goldensohn, Z. Zhang, and R.S. Raghavan, *Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization*, Proceedings of the First International Conference on Human Language Technology Research, 2001. 24

[59] Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska, *Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies*, Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4 (Stroudsburg, PA, USA), NAACL-ANLP-AutoSum '00, Association for Computational Linguistics, 2000, pp. 21–30. 4, 12

[60] Dragomir R. Radev, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Çelebi, Danyu Liu, and Elliott Drabek, *Evaluation challenges in large-scale document summarization*, Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (Stroudsburg, PA, USA), ACL '03, Association for Computational Linguistics, 2003, pp. 375–382. 1

[61] DragomirR. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara, *The acl anthology network corpus*, Language Resources and Evaluation (2013), 1–26. 37

[62] Delip Rao and Deepak Ravichandran, *Semi-supervised polarity lexicon induction*, Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (Stroudsburg, PA, USA), EACL '09, Association for Computational Linguistics, 2009, pp. 675–682. 65

[63] G. J. Rath, A. Resnick, and T. R. Savage, *The formation of abstracts by the selection of sentences. Part I. Sentence selection by men and machines*, 12 (1961), 139–141+. 24

[64] Gerard Salton and Michael J. McGill, *Introduction to modern information retrieval*, McGraw-Hill, Inc., New York, NY, USA, 1986. 15

[65] Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley, *Automatic text structuring and summarization*, Inf. Process. Manage. 33 (1997), no. 2, 193–207. 24

[66] C. Spearman, *The proof and measurement of association between two things*, The American Journal of Psychology 15 (1904), no. 1, pp. 72–101 (English). 31

[67] Xiaojun Wan and Jianguo Xiao, *Single document keyphrase extraction using neighborhood knowledge.*, AAAI (Dieter Fox and Carla P. Gomes, eds.), AAAI Press, 2008, pp. 855–860. 37

[68] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann, *Recognizing contextual polarity in phrase-level sentiment analysis*, Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (Stroudsburg, PA, USA), HLT '05, Association for Computational Linguistics, 2005, pp. 347–354. 65

[69] ______, *Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis*, Comput. Linguist. 35 (2009), no. 3, 399–433. 65

[70] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning, *Kea: Practical automatic keyphrase extraction*, Proceedings of the Fourth ACM Conference on Digital Libraries (New York, NY, USA), DL '99, ACM, 1999, pp. 254–255. 1

[71] Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng, *Text summarization using a trainable summarizer and latent semantic analysis*, Inf. Process. Manage. 41 (2005), no. 1, 75–95. 3, 14

[72] Torsten Zesch and Iryna Gurevych, *Approximate matching for evaluating keyphrase extrac-*

tion, Recent Advances in Natural Language Processing, RANLP 2009, 14-16 September, 2009, Borovets, Bulgaria, 2009, pp. 484–489. 15, 39

[73] Hongyuan Zha, *Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering*, Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '02, ACM, 2002, pp. 113–120. 1

[74] Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy, *Paraeval: Using paraphrases to evaluate summaries automatically*, Association for Computational Linguistics, April 2006. 25