Graph based Approaches for Designing Evaluation Techniques of Automatic Summarization & Keyphrase Extraction

Fahmida Hamid

Contents

List of Figures

List of Tables

Acknowledgements

Abstract

Chapter 1

Introduction

Information retrieval from text data is a challenging area of study. Automatic summarization and keyphrase extraction are two information retrieval tasks who incorporate many important aspects of both natural language understanding and natural language generation [43]. They 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 they use external knowledge sources to improve their results. Interestingly, it is quite common that a native speaker of any particular language is more efficient/comfortable on these tasks than a non-native speaker. Therefore, it is easily understood how difficult the task would be for a machine/system. *With the exponential growth-rate of data, on the other hand, performing these tasks with human-labors are absolutely expensive, time consuming; in short, 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 [25]. The web search engines, for example, have exploited the use of text summarization from the very beginning: starting with the extraction of certain number of bytes to the more sophisticated query focused summaries typified by Google's snippet [52]. 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 [49].

Automatic summarization is a more diverse problem than keyphrase extraction/generation. We can categorize the summaries in various ways based on different aspects/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 less ambiguous terms, and idiosyncratic words. No such assumption can be made for the generic summary. A *fluent* summary is written using wellformed grammatical sentences which are related and follow one another according to the rules of *coherent discourse structure*. A *disfluent* 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 where as an *indicative* one merely includes the content except 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 from automatic text summarization articles such as [12, 8].

Keyphrases/summaries can either be *extractive* or *abstractive*. The extractive approaches select/rank the salient pisces from the original source and concatenate them to yield a shorter version. An abstractive approach, on the otherhand, paraphrases using more generic terms. A person with moderate control over the language is expected to generate a set of good key-phrases (or, write an informative and meaningful summary). Since, a natural language changes over time/horizon, has innumerable ways to express a statement/sentiment, it is very difficult for a system to generate a synopsis of the article(s) like human written ones. Language generation (at least paraphrasing, sentence-fusion, word-replacement) without introducing *ambiguity* is one of the most challenging problems so far. This is the major reason for extractive approaches being more popular with systems than the abstractive ones. Summaries (and keyphrases) can be generated from *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 source-documents. Working with multi-documents has some issues (e.g., removing redundancy, maintaining the flow of information, coving all/dominant topic(s), etc.) which are not so intense with single documents.

Supervised and *unsupervised*, both 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 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/references*. It is even strenuous (and expensive) to generate/produce a standard the training set with sufficient references. Unsupervised approaches are less susceptible to the features/domains but their evaluation-score still depends on the quality of human provided references.

Many text summarization methods are surveyed in [25, 23] and in Document Understanding Conference (which is known as Text Analysis Conference since 2008). From the established and well-known approaches, we can outline some important features/techniques that are helpful in almost all the cases. Maybury et. al [23] 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 combine these features to yield a *salience* *function* that measures the significance of information [50]. Entity level approaches use different entities (word, sentence, paragraph, etc.) and their relationships (e.g., co-occurence, co-reference) to model the text and extract/rank 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 some specific sections like *conclusion, introduction, result-analysis*.

A lot of work is undergoing to leverage the performance of the systems. Hovy and Lin [12], for example, attempted to create a summarization system based on the equation: $summarization = \textit{topic identification} + interpretation + generation$. Mihalcea and Tarau [30] have proposed a graph-based ranking model for text processing, and showed how this model can be successfully used in natural language applications (specially in summarization, and keyphrase extraction). Erkan et al. [5] introduced a stochastic graph-based method for computing relative importance of the textual units for natural language processing. Radev et al. [42] 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 technique to detect *cross sentence information subsumption* to control the redundant information. Carbonell [3] 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 believe, designing a common framework to evaluating the performances of different systems is also a very crucial issue. For example, author assigned keyphrases of a research article differ to some extent from the reader (reviewer) assigned ones. In case of summarization, a human can generate slightly different summaries for the same document, if asked to do the job at different times (order). We use some absolute scales (precision, recall, f-score) or their variants to evaluate the system performances. If multiple gold-standards are available, the usual approach is to combine all of them, and find the average overlap. We suggest that *a relativized scale* could be a better alternative. In case of multiple gold standards, the references should be weighted by comparing the relatedness (closeness) among themselves. Hence, we will be proposing an evaluation technique that *evaluates the evaluators, normalize the performance of each pair (system, and reference), per task and then sum-up (and average, if needed) the score to produce the overall rank. Our proposed approach can evaluate two performances if the length of their output are not equal.* Thus, it makes the performance scores comparable between different systems (or the same system, with different set of outputs).

Research Goal

Automatic summarization and key-phrase extraction are two well known approaches to distill the most important information from a set of sources. Now a days, with the expanding nature of web, summarization and key-phrase extraction from documents based on related topics (or, questions, 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 been used to disambiguate some terms and relate one with the other. Despite their countless efforts, some results are far away from passing the turing test. *We find it interesting and necessary to expand existing algorithms aiming some performance improvement, and at the same time, to re-design the existing evaluation techniques to consider some facts*.

First, we plan to exploit different graph based techniques to leverage the performance of the tasks. For example, we reuse a graph-based algorithm (initially designed to symbolize a trust based network) to extract sentiment-polar summaries from each document. We also have a framework based on the same model to automatically generate context-aware sentiment-lexicon per topic. We expand the topic-based textrank algorithm with some context information to extract topic-focused key-phrases. Even using a lot of external information, we have imperfect outcomes. Hence, we start using some simple techniques to *re-generate*, i.e., *deform/twist* the results we have from *extractive* approaches. Here, we find the clue to our next research-topic. While humans, by nature, choose abstractive outputs over extractive ones, most of the systems follow extractive algorithms. So *there is always 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.*

Manually labeling the datasets, and generating base-cases for comparison is infeasible, and time consuming. We need some minimal manual outputs to evaluate the performance of the systems. 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. Hence, *to weigh gold standards with some discrepancy* is a challenge to the evaluation approaches. Evaluation based on fixed points and averaging are an old style. We re-define the *absolute* scales with a more *flexible relativized approach*, come up with a mathematically sound *baseline generation* technique. *We show how to evaluate the evaluators and thus improvise a scaling mechanism for evaluating summaries and key-phrases*.

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 [24]. An intrinsic (or normative) evaluation judges the quality of the summary directly based on analysis in terms of some 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. *It is noticeable that a summary created with such high compression ration is hardly distinguishable from a set of keyphrases. It usually does not contain syntactically correct sentences, but some phrases connected through punctuations.* 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*. Also, 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 develop an automatic evaluation technique that can adjust to the length variation.

We can outline of our research questions (and goals) with the following statements:

- Design a Fair Evaluation Technique for Automatic Summarization & Keyphrase Extraction that possesses the following properties
  - a mathematically sound baseline
  - a relativized scale that takes into account different outputs with possibly different lengths
  - a methodology to define the degree of agreement (confidence) between the human evaluators
  - embed a semantic knowledge-base with the evaluation process so that the extractive and the abstractive approaches become comparable with each other
  - check syntatic well-formedness
- Create a large database and at least one reliable reference (standard) set that can be useful to both of the tasks

Besides working on the *evaluation track*, we also have employed our attention to designing a sentiment oriented summarization model [11], and have presented the usefulness of summarization as an assistive technology to the people with vision deficiencies [10].

Chapter 2

Evaluation Techniques

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 [8].

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 [15]. 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. *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* [16]. 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 aspects: *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/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 [41]. 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.

$$precision = \frac{\text{relevant information present in the system summary}}{\text{total information extracted by the system summary}}$$

$$recall = \frac{\text{relevant information present in the system summary}}{\text{total information present in the ideal summary}}$$

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

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

Other common measures include $Kappa$ [4] and relative utility [42]. 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, 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)*.

While evaluating with an *absolute scale*, Kappa, CBSU, 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 ($i\text{-}measure$) [9] with some weighted-matching strategy that is suitable for evaluating both of the tasks. We, then, propose a modified approach of $i\text{-}measure$ that not only can adjust to the length variation but also considers *an equivalence-relation* using WordNet provided *Synsets* to modify the *observed intersection (matching) size*.

2.2 Related Work

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 [50].

However, most IR 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 [12].

ROUGE [19] is one of the well-known techniques to evaluate single/multi-document summaries. ROUGE includes measures to automatically determine the quality 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.

Another summary evaluation tool *Pyramid* [34] 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*. But the *SCUs were defined by humans, which is a restriction to designing a completely automatic evaluation tool.*

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* value on multiple points (such as for top $5, 10, 15$ keyphrases) to report their system's performance. If multiple references are available, there is even no proper definition of how to weigh their agreement/disagreement. Another important issue is, the reference set does not always contain exactly same number of keyphrases as the system generated sets. *R-p* [51] (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 [45]. In information retrieval, *R-p* is the *precision* when the number of retrieved documents equals the number of relevant documents.

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/thesaurus for comparison. The evaluation strategy should be strong enough to compare the systems/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/ key-phrases, 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/key-phrases 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 2.

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

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 $I = L \cap K$.

$$\begin{aligned} \Pr(x \in K) \cdot \Pr(x \in L) &= \Pr(x \in (L \cap K)) \\ &= \Pr(x \in I) \end{aligned} \tag{2.1}$$

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 eq. 4.1 we deduce,

$$\begin{aligned} (k/n)(l/n) &= i/n \\ i &= \frac{kl}{n} \end{aligned} \tag{2.2}$$

A similar idea was stated briefly by Goldstein [7], but is not brought to much usage for evaluation purpose.

2.4 The 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*:

$$\begin{aligned} i\text{-}measure(N, K, L) &= \tfrac{observed\_size\_of\_intersection}{expected\_size\_of\_intersection} \\ &= \tfrac{|K \cap L|}{\frac{|K| \cdot |L|}{|N|}} = \tfrac{\omega}{\left(\frac{kl}{n}\right)} = \tfrac{\omega}{i} \end{aligned} \tag{2.3}$$

2.4.1 Connect Relativized Scale to Absolute Scale

Recall, Precision, and*f-measure* are the re-known 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 *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 eqn 4.2 as:

$$\begin{aligned} f\text{-}measure_{expected} &= 2pr/(p+r) \\ &= \tfrac{2 \cdot i^2}{k \cdot l} / \tfrac{(k+l) \cdot i}{k \cdot l} \\ &= 2i/(k+l) \\ &= i/((l+k)/2) \end{aligned} \tag{2.4}$$

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,

$$\begin{aligned} f\text{-}measure_{observed} &= 2pr/(p+r) \\ &= \tfrac{2 \cdot \omega^2}{k \cdot l} / \tfrac{(k+l) \cdot \omega}{k \cdot l} \\ &= 2\omega/(k+l) \\ &= \omega/((k+l)/2) \end{aligned} \tag{2.5}$$

By substituting ω and i using eq.4.5 and 4.4, we get,

$$i\text{-}measure(N, K, L) = \frac{f\text{-}measure_{observed}}{f\text{-}measure_{expected}} \tag{2.6}$$

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: Sample Cases: *i-measure*

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

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

Table 2.2: Author-written Abstract & TextRank Output

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 4.2 shows that $|K| = 6$, $|L| = 15$, and $|N|$ was counted as 849. The average-size of intersection, i is $\frac{6*15}{849}$, which is 0.016. And there is 2 exact match, i.e. $\omega = 2$. Therefore, $i\text{-}measure = \frac{2}{0.016}$, i.e., 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.

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/ compare the performance between different systems. Our major concern is, in order to compare the performance of between different system-outputs, they should be tested several times against each-other. 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 *words/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.

Chapter 3

Evaluation Technique with Multiple References

3.1 Introduction

Human quality text summarization systems are difficult to design and even more difficult to evaluate [7]. 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. Work is also on going to improve anaphora resolution, 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 [39]. 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 [20].

Several studies ([46], [44], [27], [25]) suggest that *multiple human gold-standard summaries would provide a better ground for comparison*. Lin [18] states that multiple references tend to increase evaluation stability although human judgements only refer to single reference summary.

After considering the evaluation procedures of ROUGE [19], Pyramid [35], and their variants e.g., ParaEval [53], 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 3.3). Such a strategy was discussed briefly by Goldstein et al. [7]. 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

Most of 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* [7].

ROUGE (Recall Oriented Understudy for Gisting Evaluation) [19] is one of the well known techniques to evaluate single/multi-document summaries. ROUGE is closely modelled after BLEU [38], a package for machine translation evaluation. ROUGE includes measures to automatically determine the quality 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.

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

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 [13].

Nenkova [35] 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

Another piece of work that we would like to mention is the Pyramid method [34]. A key assumption of the method is the need for multiple models, which taken together, yield a gold standard for system output. A pyramid represents the opinions of multiple human summary writers each of whom has written a model summary for the multiple set of documents. Each tier of the pyramid *quantitively* represents the agreements among human summaries based on *Summary Content Units (SCU)* which are content units, 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*. 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 -

- Pyramid method does not define a *baseline* to compare the degree of (dis)agreement between human summaries.
- High frequency units receive higher weights in the Pyramid method. Nenkova [36], 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.
- There is no clear direction about the summary length (or compression ratio).

Our method uses a unique baseline for all (system, and reference summaries) and it does not need the absolute scale (like f, p, r) to score the summaries.

3.3 Evaluating a System's Performance with Multiple References

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 ones. 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 & System

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-measure* between $\binom{m}{2}$ pairs of reference summaries and normalize with respect to the best pair. We also compute the *i-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

$$\begin{array}{rcl} w_d(h_p, h_q) & = & \frac{i\text{-}measure(d, h_p, h_q)}{\mu_d} \\ w_d(s_j, h_p) & = & \frac{i\text{-}measure(d, s_j, h_p)}{\mu_{(d,h_p)}} \end{array} \quad (3.1)$$

where,

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

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

Table 3.2: normalized i-measure of all possible reference pairs for document: $D30053.APW19981213.0224$

$Pair(p,q)$	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

3.3.2 Confidence based 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:

$$c_d(h_p) = \frac{\sum_{q=1,p\neq q}^{m}(w_d(h_p,h_q))}{m-1}. \tag{3.2}$$

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

$$score(s_j,d) = \sum_{p=1}^{m} c_d(h_p) \times w_d(s_j,h_p) \tag{3.3}$$

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

$$i\text{-}score(s_j) = \frac{\sum_{i=1}^{t} score(s_j,d_i)}{t}. \tag{3.4}$$

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 summaries (B,G,E,F) and three machine summaries on document $D30053.APW19981213.0224$

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

3.4 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

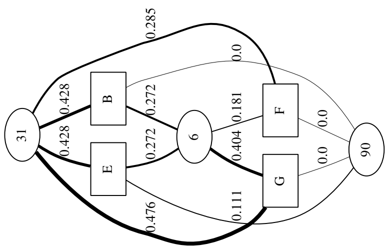


Figure 3.1: system-reference graph: edge-weights represent the normalized $i\text{-}measure$

Table 3.4: Confidence based System Score

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

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 3.3. Then we use formula 3.4 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.

3.5 Experimental Results

We perform different experiments over the dataset. Section 3.5.1 describes how $i\text{-}measure$ among the reference summaries can be used to find the confidence/ nearness/ similarity of judgements. In section 3.5.2, we examine two types of rank-correlations (pair-based, distance based) generated by $i\text{-}score$ and $ROUGE\text{-}1$. Section 3.5.3 states the correlation of $i\text{-}measure$ based ranks with human assessors.

3.5.1 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 3.5) 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

Table 3.5: normalized i-measure of all possible reference pairs for $D30015.APW19981005.1082$

$Pair(p,q)$	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 3.6: Confidence Score

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

Table 3.7: Rank Correlations

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 3.8: guess score for $D188$, assessor F

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

their confidence score(table 3.3) is fairly high. But table 3.6 shows that most of the references do not share common words, hence $confidence$ values of the references for document $D30015.APW19981005.1082$ is quite different from each other.

3.5.2 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 [47] and *Kendall's* τ [17], (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/discordant pairs (Table 3.7). 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.

3.5.3 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 3.8 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

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

3.5.4 Critical Discussion

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/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/word) might be a trigger. As $i\text{-}measure$ is lenient on lengths, we can modify eq. 3.3 with the following to apply *extraction/generation of proper sentences* within a maximum word/sentence window as an impact factor.

$$score(s_j, d) = \left(\sum_{p=1}^{m} c_d(h_p) \times w_d(s_j, h_p)\right) \times \frac{c\_sen}{t\_sen} \tag{3.5}$$

where, t_sen is the total number of sentences produced/ 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/phrases* model (e.g., $i\text{-}measure$) can be applied over it and a score can be produced using eq. 3.5.

Standard sentence tokenizers, POS taggers, etc. can be used to analyze sentences. The word/ 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.

3.6 Conclusion

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/ topic-specific summary evaluation task.

Chapter 4

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 [28].

4.1 Introduction

Automatic Summarization and Keyphrase Extraction are two widely used areas in Natural Language Processing and Information Retrieval. A reliable dataset is extremely helpful to evaluate the systems performing these tasks. 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 contribution, in this work, is twofold: firstly, we provide a dataset with a highly trustable reference set of summaries and keywords, and secondly, we provide an evaluation mechanism for systems that are mostly extractive whereas the references are, by nature, abstractive.

The scientific articles come with author-assigned *abstracts* and *keyphrases* that provide highest-possible quality *references* to be used as a baseline for the evaluation of system generated ones, thus 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.

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. Our effort in this work 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 an evaluation methodology that can adjust to the length variation between reference set(s) and the system generated outputs, and also use a *weighted relatedness* when comparing between two groups of output.

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 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, i.e., is not a good idea to have one's favorite category-theory, genomics, or string-theory paper summarized by the 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 the 1250 test documents, 1218 of the abstracts contain words (excluding stopwords/punctuations) which are not present in the original document. And, 1186 of them contain words in the author-provided keyphrases that could not be found in the original article. So, some form of *semantic equivalence relation* is needed, instead of word phrase equality to compare fairly. 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 (e.g., precision, recall, f-measure) to a relativized approach, e.g., $i\text{-}measure$ [9].

In order to show the usage of our dataset, we produce some candidate-output using the classic unsupervised algorithm: TextRank [30]. 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 [14] comes with 2000 abstracts as an attempt to enhance the performance automatic keyword extraction. The SP dataset [37] contains around 120 scientific articles with *author-assigned* and *reader-assigned* keyphrases. The DUC dataset [48] contains 308 documents from DUC2001, manually annotated by two indexers. 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) [40] 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 [20]. Besides, it is *time-consuming and very expensive* to produce human annotated reference set(s) for a large collection.

4.3 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. *ROUGE* [19] is one of the well-known techniques to evaluate single/multi-document summaries. ROUGE includes measures to automatically determine the quality 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.

Another summary evaluation tool *Pyramid* [34] 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*. But the *SCUs were defined by humans, which is a restriction to designing a completely automatic evaluation tool*.

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* value on multiple points (such as for top $5, 10, 15$ keyphrases) to report their system's performance. If multiple references are available, there is even no proper definition of how to weigh their agreement/disagreement. Another important issue is, the reference set does not always contain exactly same number of keyphrases as the system generated sets. *R-p* [51] (a deviation from pure precision) was proposed by authors to consider various lengths of outputs together.

Considering these 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*. We, therefore, propose a relativized scale with some weighted-matching strategy that is suitable for evaluating both of the tasks.

4.4 The Relativized Scale

While evaluating with an *absolute scale*, 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. We propose a modified approach of $i\text{-}measure$ [9] 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*.

We will be describing the baseline generation scenario briefly, and then explain how we relate the system generated abstractive/extractive approaches with the provided references.

4.4.1 A Sound Baseline for All

We consider each document (and generated summary/ key-phrases, 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/key-phrases 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 2.

Hypothesis 2 *The size of overlap between two sets of output should be compared against the average intersection size of two random sets.*

4.4.2 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 $I = L \cap K$.

$$\begin{aligned} \Pr(x \in K) \cdot \Pr(x \in L) &= \Pr(x \in (L \cap K)) \\ &= \Pr(x \in I) \end{aligned} \tag{4.1}$$

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 eq. 4.1 we deduce,

$$\begin{aligned} (k/n)(l/n) &= i/n \\ i &= \frac{kl}{n} \end{aligned} \tag{4.2}$$

A similar idea was stated briefly by Goldstein [7], but is not brought to much usage for evaluation purpose.

4.4.3 The 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*:

$$\begin{aligned} i\text{-}measure(N, K, L) &= \tfrac{observed\_size\_of\_intersection}{expected\_size\_of\_intersection} \\ &= \tfrac{|K \cap L|}{\frac{|K| \cdot |L|}{|N|}} = \tfrac{\omega}{\left(\frac{kl}{n}\right)} = \tfrac{\omega}{i} \end{aligned} \tag{4.3}$$

4.4.4 Connect Relativized Scale to Absolute Scale

Recall, Precision, and*f-measure* are the re-known 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 *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 eqn 4.2 as:

$$f\text{-}measure_{expected} = i/((l + k)/2) \tag{4.4}$$

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,

$$f\text{-}measure_{observed} = \omega/((k + l)/2) \tag{4.5}$$

By substituting ω and i using eq.4.5 and 4.4, we get,

$$i\text{-}measure(N, K, L) = \frac{f\text{-}measure_{observed}}{f\text{-}measure_{expected}} \tag{4.6}$$

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.*

4.5 Adapt Abstractiveness

After exploring the existing automatic summarization and keyphrase extraction algorithms, we have discovered that the majority of the algorithms focuses on *extractive* methods. Most 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 or she produces often new sentences / or keyphrases, possibly not occurring as such in the document. Therefore, our hypothesis, that *the reference-set is a complete subset of the article ($K \subseteq N$) is not always true*.

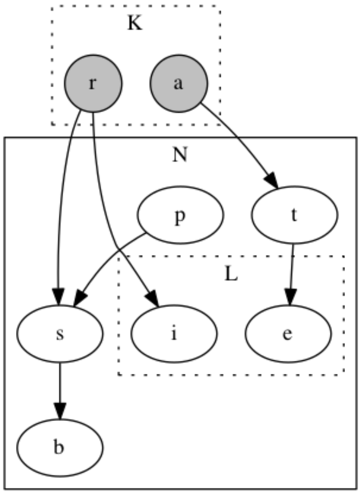


Figure 4.1: words in N, L, and K are related

In such circumstances, we should deviate from the exact string/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 maningful, the semantic relations need to be relativized to an ontology. As the first approximation we use WordNet to find synset based similarity for evaluation but a richer, more knowledge intensive source like Wikipedia is planned as a future development.

The Equivalence Class

WordNet [32] 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/word matching approach.

Table 4.1: Synset-based Similarity Measure

Words in Abstract: K	Synset
data	{(’datum.n.01’), (’data.n.01’)}
TextRank Generated Summary: L	Synset
information	{(’information.n.02’), (’information.n.01’), (’information.n.05’), (’data.n.01’), (’information.n.03’)}

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

4.6 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 eq. 4.3 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 [9].

4.7 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/abc/dataset_ictir`, free for research purposes. [1]

4.8 Case Study

In order to explain the usefulness of the evaluation tool, we pick the keyphrase extraction task and a sample document as our case study. We show why/how the evaluation tool is more effective than the absolute scale.

Table 4.2 shows that $|K| = 6$, $|L| = 15$, and $|N|$ was counted as 849. The average-size of intersection, i is $\frac{6*15}{849}$, which is 0.016. And there is 2 exact match, i.e. $\omega = 2$. Therefore, $i\text{-}measure = \frac{2}{0.016}$, i.e., 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.

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 *words/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.

[1] The original link is distorted due to blind review issue

Table 4.2: Diamond Standard & TextRank Output

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

4.9 Conclusion

Accurate evaluation is useful - including for their use in machine learning. Tools like the $i\text{-}measure$ introduce some flexibility. Our version of $i\text{-}measure$ 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 and paraphrase detection techniques. We believe, it will be a great resource to the research community who are working with information extraction from scientific articles.

Chapter 5

Embedding Knowledge-bases for Evaluation

Texts are not simply a flat list of sentences; they have a hierarchical structure, one in which certain clauses are more important than others [12].

Chapter 6

Summarization Model focused on Sentiment

We propose an *unsupervised* model to extract two types of summaries *(positive, and negative)* per document based on sentiment polarity. Our model builds a *weighted polar digraph* from the text, then evolves recursively until some desired properties converge. It can be seen as an enhanced variant of *TextRank* type algorithms working with non-polar text graphs. Each positive, negative, and objective opinion has some impact on the other if they are semantically related or placed close in the document.

Our experiments cover several interesting scenarios. In case of a one author news article, we notice a significant overlap between the *anti-summary* (focusing on negatively polarized sentences) and the the summary. For a transcript of a debate or a talk-show, an anti-summary represents the disagreement of the participants on stated topic(s) whereas the summary becomes the collection of positive feedbacks. In this case, the anti-summary tends to be *disjoint* from the regular summary. Overall, our experiments show that our model can be used with TextRank to enhance the quality of the extractive summarization process.

6.1 Introduction

A document expresses a writer's opinions along with 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 amongst 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 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. Besides, 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. Thus we build a *weighted directed graph* and apply our model to get the top (positively 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/ 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* 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).

6.2 Background Study

Automated text summarization dates back to the end of fifties [22]. 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 [7].

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 [7] 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 [12] 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.

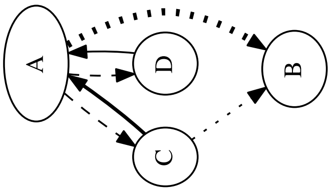


Figure 6.1: A text-graph describing several topics

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 [30]. 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 [5] 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.

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* [33]. 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.

6.3 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, …*$\}$, 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/ 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, anti-summaries are not exactly opposite to summaries, 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 summaries and the other's anti-summaries.

6.4 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 [21]. 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.

6.5 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 [30]. 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 [29]. 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 [33] that can be adopted to have the impact of sentiments on the summary.

6.5.1 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 [33] correlated two attributes for each node: *Bias* and *Prestige*.

Bias & 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

$$bias(i) = \frac{1}{2|d^o(i)|} \sum_{j \in d^o(i)} (w_{ij} - rank(j)) \tag{6.1}$$

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

$$rank(j) = \frac{1}{|d^i(j)|} \sum_{k \in d^i(j)} (w_{kj}(1 - X_{kj})) \tag{6.2}$$

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

$$X_{kj} = \begin{cases} 0 & \textit{if}\,(bias(k) \times w_{kj}) \leq 0, \\ |bias(k)| & \textit{otherwise}. \end{cases} \tag{6.3}$$

After each iteration of 6.1 and 6.2, edge weight w_{kj} is scaled from the old weight as follows:

$$w_{kj}^{new} = w_{kj}^{old}(1 - X_{kj}) \tag{6.4}$$

6.6 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 [6] assigns to each synset of *WordNet* [31] 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: Build 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: Apply Ranking Model
  - Apply formula [6.1,6.2,6.3,6.4] over the graph until the $rank$ value converges.
- Phase III: Find Ranked Word Vectors, & Extract 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: `http://students.cse.unt.edu/~fh0054/SummaryAnti/Kennedy1961/kennedyPart1.txt`, which is a small fragment (only 77 sentences are considered) of President Kennedy's speech in 1961.

6.6.1 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 6.1, 6.2 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.

6.6.2 Define Nodes & 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.

6.6.3 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,

$$similarity(x, y) = \frac{def(x) \cap def(y)}{def(x) \cup def(y)} \tag{6.5}$$

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

Table 6.1: 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 6.2: 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

- 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 6.2. We notice that word pairs $\{$*(security, recession), (security, recovery), (progress, recovery)*$\}$, for example, are connected to each other through the *similarity* relationship.

6.6.4 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 [6.1,6.2,6.3,6.4] several times (until the rank value converges) over the graph. For real time output, one can control the repetition using a threshold. Table 6.3 shows a handful of positively ranked and negatively ranked keywords (out of 568 total words) from the same article.

6.6.5 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

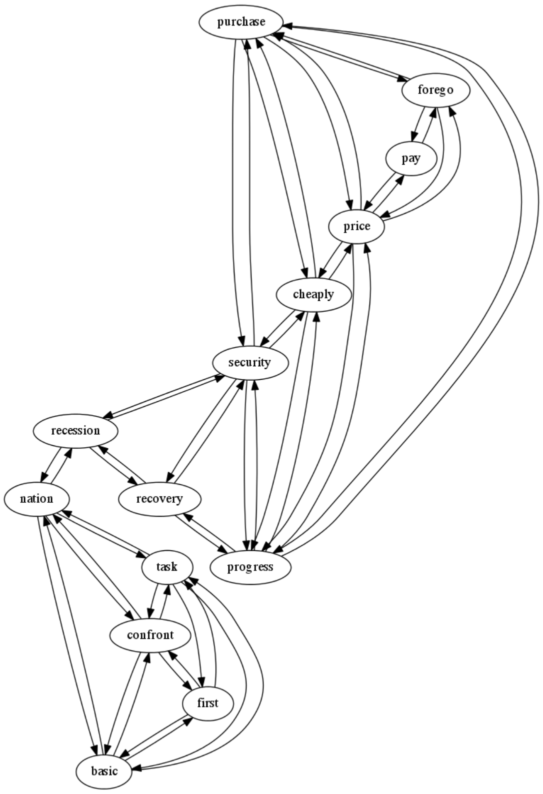


Figure 6.2: Sentence Graph

Table 6.3: 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	

of a sentence, s_j is:

$$weight(s_j^{neg}) = (\sum_{v_i \in W^{neg} \wedge v_i \in s_j} |rank(v_i)|)/(n \times |s_j|), \tag{6.6a}$$

$$weight(s_j^{pos}) = (\sum_{v_i \in W^{pos} \wedge v_i \in s_j} rank(v_i))/(m \times |s_j|), \tag{6.6b}$$

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 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 6.4 shows two top sentences(the first one is the 2^{nd} top positively ranked and the second one is 5^{th} top negatively ranked). The original file can be found at: `http://students.cse.unt.edu/~fh0054/SummaryAnti/Kennedy1961/kennedyPart1SA.txt`.

Table 6.4: 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

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.

6.7 Evaluation

We used TextRank (extracted) and Human (abstract) summaries from DUC 2004 (task 1) as the baseline. 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 6.5 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 these set of articles are stored in link: `http://students.cse.unt.edu/~fh0054/cicling2015/`). Quite interestingly, for shorter articles, *case a* and *case b* showed similar (and better) performance than *case c* and *case d* [table 6.5, & figure 6.3]. It also supports hypothesis 1. For larger articles, *case b* was the winner. The following section gives the mathematical background for case d.

Table 6.5: summary & their average size of intersection

Test Set	Total Files	no of sentences per file(n)	$avg(n)$	summary size(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

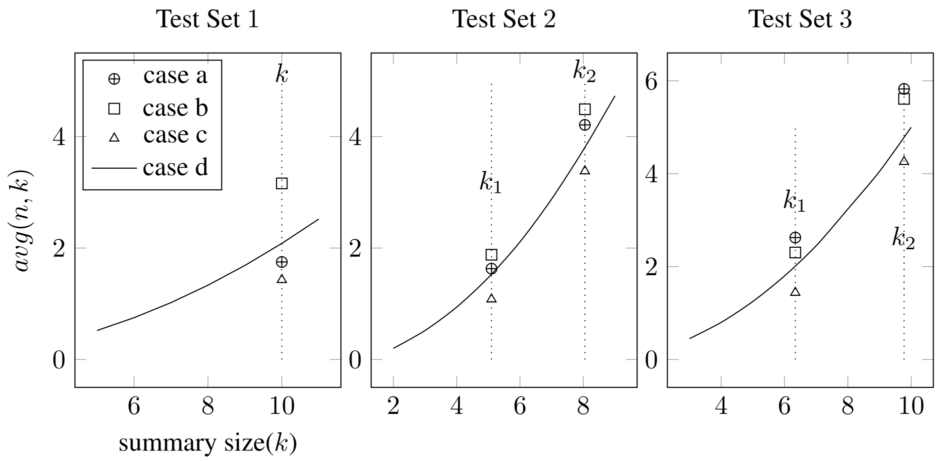


Figure 6.3: average of sentence intersection based on equation 6.7

6.7.1 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 eq. 6.7.

$$avg(n,k) = \frac{\sum_{i=0}^{k} i \binom{k}{i} \binom{n-k}{k-i}}{\sum_{i=0}^{k} \binom{k}{i} \binom{n-k}{k-i}} \tag{6.7}$$

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

$$avg(n,k,l) = \frac{\sum_{i=0}^{k} i\binom{k}{i}\binom{n-k}{l-i}}{\sum_{i=0}^{k} \binom{k}{i}\binom{n-k}{l-i}} \tag{6.8}$$

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.

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.

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

In each case, $(P \cap N)$ is minimal (fig. 6.3) 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?

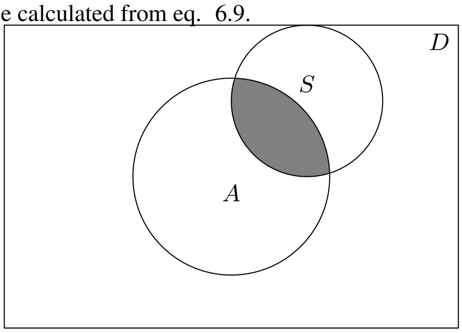
We tried to answer these questions based on experimental results. We might need voluntary human judges to label the articles based on the the extractive summaries and compare our *summary based guesses*. We leave this phase as a future direction. Interested reader can get our result from the following link: `http://students.cse.unt.edu/~fh0054/cicling2015/summaryHalf/fileType.txt`

6.7.3 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.

Recall & Precision

Recall (R) is the ratio of *number of relevant information received to the total number of relevant information in the system*. If D is the original document, A is the anti-summary, and S is the standard summary, then the recall(R) value can be calculated from eq. 6.9.



$$R = \frac{A \cap S}{S} \tag{6.9}$$

$$P = \frac{A \cap S}{A} \tag{6.10}$$

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 (eq. 6.10). 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 6.6: Avg. Recall of P, N, T w.r.t. Human Summary

Test Set	Total Files	no of sentences per file(n)	summary size(k sentences)	recall(P) $= \frac{P \cap H}{H}$	recall(N) $= \frac{N \cap H}{H}$	recall(T) $= \frac{T \cap H}{H}$
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 6.6 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. We have used *uni-gram 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.*

6.7.4 Evaluation through examples: $(P \cap N)$ is minimal

The next block is a summary and an anti-summary produced by our system, for the data file: `http://students.cse.unt.edu/~fh0054/SummaryAnti/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 Googles privacy policy. On the other hand, the *anti-summary* sentences are about *Googles* 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.

6.7.5 Evaluation through examples: $(P \cap N)$ is maximal

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: `http://students.cse.unt.edu/~fh0054/SummaryAnti/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[26]'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.

6.8 Conclusion

Our approach is domain independent and *unsupervised*. From our experiments we can deduce that most of the important sentences contain a good mixture of positive and negative sentiments toward related topics; a noticeable amount of extracted sentences overlap between the summary and the anti-summary for a *non-biased* article; and a sentiment oriented summary can be produced by extending TextRank summary with anti-summary model generated one.

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 polarity-direction. 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.

Chapter 7

Conclusion

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 [16].

Bibliography

[1] A. Balahur, M. A. Kabadjov, J. Steinberger, R. Steinberger, and A. Montoyo. Summarizing opinions in blog threads. In O. Kwong, editor, *PACLIC*, pages 606–613. City University of Hong Kong Press, 2009.

[2] P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. Exploring sentiment summarization. In Y. Qu, J. Shanahan, and J. Wiebe, editors, *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. AAAI Press, 2004. AAAI technical report SS-04-07.

[3] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *In SIGIR*, pages 335–336, 1998.

[4] J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.*, 22(2):249–254, June 1996.

[5] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, Dec. 2004.

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

[7] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 121–128, New York, NY, USA, 1999. ACM.

[8] U. Hahn and I. Mani. The challenges of automatic summarization. *Computer*, 33(11):29–36, Nov. 2000.

[9] F. Hamid, D. Haraburda, and P. Tarau. Evaluating text summarization systems with a fair baseline from multiple reference summaries. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, pages 351–365, 2016.

[10] F. Hamid and P. Tarau. Text summarization as an assistive technology. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2014, Island of Rhodes, Greece, May 27 - 30, 2014*, pages 60:1–60:4, 2014.

[11] F. Hamid and P. Tarau. Anti-summaries: Enhancing graph-based techniques for summary extraction with sentiment polarity. In *Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part II*, pages 375–389, 2015.

[12] E. Hovy and C.-Y. Lin. Automated text summarization and the summarist system. In *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, TIPSTER '98, pages 197–214, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[13] E. Hovy, C. yew Lin, L. Zhou, and J. Fukumoto. Automated summarization evaluation with basic elements. In *In Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC*, 2006.

[14] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[15] H. Jing, R. Barzilay, K. Mckeown, and M. Elhadad. Summarization Evaluation Methods: Experiments and Analysis. In *In AAAI Symposium on Intelligent Summarization*, 1998.

[16] K. S. Jones. Towards better nlp system evaluation. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, pages 102–107, 1994.

[17] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):pp. 81–93, 1938.

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

[19] C.-Y. Lin. Rouge: a package for automatic evaluation of summaries. pages 25–26, 2004.

[20] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[21] B. Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[22] H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, Apr. 1958.

[23] I. Mani. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA, USA, 1999.

[24] I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, and B. Sundheim. The tipster summac text summarization evaluation, 1999.

[25] I. Mani and M. T. Maybury. Automatic summarization. In *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.*, page 5, 2001.

[26] Y. Mao and G. Lebanon. Isotonic conditional random fields and local sentiment flow. In *Advances in Neural Information Processing Systems*, 2007.

[27] D. Marcu. From discourse structures to text summaries. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 82–88, 1997.

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

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

[30] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, July 2004.

[31] G. A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41, 1995.

[32] 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(4):235–244, 1990.

[33] A. Mishra and A. Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 567–576, New York, NY, USA, 2011. ACM.

[34] A. Nenkova, R. Passonneau, and K. McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2), May 2007.

[35] A. Nenkova and R. J. Passonneau. Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL*, pages 145–152, 2004.

[36] A. Nenkova and L. Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*, 2005.

[37] T. D. Nguyen and M.-Y. Kan. Key phrase extraction in scientific publications. In *Proceeding of International Conference on Asian Digital Libraries*, pages 317–326, 2007.

[38] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

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

[40] D. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara. The acl anthology network corpus. *Language Resources and Evaluation*, pages 1–26, 2013.

[41] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Computational Linguistics*, 28(4):399–408, 2002.

[42] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum '00, pages 21–30, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[43] D. R. Radev, S. Teufel, H. Saggion, W. Lam, J. Blitzer, H. Qi, A. Çelebi, D. Liu, and E. Drabek. Evaluation challenges in large-scale document summarization. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 375–382, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[44] 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. 12:139–141+, 1961.

[45] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[46] G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarization. *Inf. Process. Manage.*, 33(2):193–207, Mar. 1997.

[47] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):pp. 72–101, 1904.

[48] X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In D. Fox and C. P. Gomes, editors, *AAAI*, pages 855–860. AAAI Press, 2008.

[49] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, pages 254–255, New York, NY, USA, 1999. ACM.

[50] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng. Text summarization using a trainable summarizer and latent semantic analysis. *Inf. Process. Manage.*, 41(1):75–95, Jan. 2005.

[51] T. Zesch and I. Gurevych. Approximate matching for evaluating keyphrase extraction. In *Recent Advances in Natural Language Processing, RANLP 2009, 14-16 September, 2009, Borovets, Bulgaria*, pages 484–489, 2009.

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

[53] L. Zhou, C.-Y. Lin, D. S. Munteanu, and E. Hovy. Paraeval: Using paraphrases to evaluate summaries automatically. Association for Computational Linguistics, April 2006.