

Summarization System Evaluation Revisited: N-Gram Graphs

GEORGE GIANNAKOPOULOS and VANGELIS KARKALETSIS

National Centre for Scientific Research Demokritos

GEORGE VOUROS

University of the Aegean

and

PANAGIOTIS STAMATOPOULOS

University of Athens

This article presents a novel automatic method (AutoSummENG) for the evaluation of summarization systems, based on comparing the character n-gram graphs representation of the extracted summaries and a number of model summaries. The presented approach is language neutral, due to its statistical nature, and appears to hold a level of evaluation performance that matches and even exceeds other contemporary evaluation methods. Within this study, we measure the effectiveness of different representation methods, namely, word and character n-gram graph and histogram, different n-gram neighborhood indication methods as well as different comparison methods between the supplied representations. A theory for the a priori determination of the methods' parameters along with supporting experiments concludes the study to provide a complete alternative to existing methods concerning the automatic summary system evaluation process.

Categories and Subject Descriptors: I.2.7 [Artificial Intelligence]: Natural Language Processing—Text analysis, language models

General Terms: Algorithms, Languages, Measurement, Performance

Additional Key Words and Phrases: Automatic summarization, summarization evaluation, n-gram graph

The research described within this article was supported by the research and development project ONTOSUM (<http://www.ontosum.org>), which is in turn funded by the Greek General Secretariat for Research and Technology.

Authors' addresses: G. Giannakopoulos and V. Karkaletsis, Institute of Informatics and Telecommunication, National Centre for Scientific Research, Demokritos, Greece; email: ggianna@iit.demokritos.gr; G. Vouros, Ai-Lab-Department of Information and Communication Systems Engineering, University of the Aegean; P. Stamatopoulos, Department of Informatics and Telecommunications, University of Athens.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org. © 2008 ACM 1550-0875/2008/10-ART5 \$5.00 DOI 10.1145/1410358.1410359 <http://doi.acm.org/10.1145/1410358.1410359>

ACM Transactions on Speech and Language Processing, Vol. 5, No. 3, Article 5, Publication date: October 2008.

ACM Reference Format:

Giannakopoulos, G., Karkaletsis, V., Vouros, G., and Stamatopoulos, P. 2008. Summarization system evaluation revisited: N-gram graphs. *ACM Trans. Speech Lang. Process.* 5, 3, Article 5 (October 2008), 39 pages. DOI = 10.1145/1410358.1410359 <http://doi.acm.org/10.1145/1410358.1410359>

1. INTRODUCTION

The overinformation commonground of recent information retrieval efforts has created a serious motive for the design and implementation of summarization systems, which are either based on existing information retrieval practices or provide a new point-of-view on the retrieval process. The value of summarization has been identified in the domain of cognitive psychology as a means of optimizing learning [Anderson 1985, p. 14]. The efforts for automatic extraction of summaries date back to 1958 and Luhn [1958] but is still an open issue. The difficulty in the automation of the summarization process is that summarization, especially from multiple documents, proves to be an abstractive mental process [Dang 2005], which is not easy to model (see also Endres-Niggemeyer [2000]). However, several systems have managed to extract summaries that are rather informative even though they seem to suffer in terms of the legibility of the summary text [Dang 2005]. On the other hand, the evaluation of summarization systems seems to be nontrivial in itself.

1.1 Motivation

We focus on the problem of evaluating summarization systems in an automated fashion. In recent scientific attempts to evaluate summarization systems, a multitude of problems arose, concerning mostly the interjudge disagreement as well as the difficulty of automatically determining the quality of a summary. These problems are found mainly in the domain of multidocument summarization where the synthesis of summaries appears to be more than mere extraction of text snippets [van Halteren and Teufel 2003; Nenkova 2006].

The problem of interjudge disagreement, as indicated in van Halteren and Teufel [2003] and Lin and Hovy [2002], is the result of human subjectivity in terms of evaluation of summaries. It has been noted that human judges appointed to grade the quality of a summary disagree notably with each other on the grades assigned to different summaries. Several methodologies have been examined to systematize the grade-assignment process, aiming to smooth or nullify the disagreement caused by methodology-specific practices [Nenkova 2006; Radev et al. 2000; Marcu 2000; Saggion and Lapalme 2002]. If one fails to create a methodology for humans to correlate vigorously with each other in the process of evaluation, then either the process of evaluation cannot be modeled objectively, which would be interesting to examine further by itself, or we need to define the process of summarization and its evaluation more precisely (for a more thorough discussion see Giannakopoulos et al. [2006]).

On the other hand, the rankings posed by human grading over summarization systems correlate strongly. This indicates that people tend to prefer the same systems over other systems, which leads to research concerning automatic

evaluation methods that produce similar rankings as we will shortly present. In order to achieve the ranking, several attributes of summarization systems have to be examined and graded. These attributes are based on the qualities of the output summaries.

The problem of automatically determining the quality of a given summary can be approached using two different perspectives: either by *intrinsic* or *extrinsic* evaluation [Mani and Bloedorn 1997; van Halteren and Teufel 2003]. Intrinsic evaluation operates on the characteristics of the summary itself, independent of the domain that may be used, trying, for example, to capture how many of the ideas expressed in the original sources appear in the output. On the other hand, extrinsic evaluation decides upon the quality of a summary depending on the effectiveness of using the domain for a specified task. Such a measure of extrinsic evaluation, namely, *responsiveness*, appeared in the Document Understanding Conference (DUC) 2005.¹ This extrinsic measure has been used in later DUCs as well.

In DUC 2005, the appointed task was the synthesis of a 250-word, well-organized answer to a complex question where the data of the answer would originate from 25 documents [Dang 2005]. In DUC 2005, the question that the summarizing peers, that is, summarizer systems or humans, were supposed to answer consisted of a topic identifier, a title, a narrative question and a granularity indication, with values ranging from general to specific. The responsiveness score was supposed to provide, as Dang [2005] states, a “coarse ranking of the summaries for each topic, according to the amount of information in the summary that helps to satisfy the information need expressed in the topic statement, at the level of granularity requested in the user profile”. In other words, the responsiveness grade was meant to result in a partial ordering indicative of how well a given summary answers a given question taking into account the specifications of a question. According to the definition of extrinsic evaluation, responsiveness is an extrinsic evaluation measure.

The responsiveness grade was appointed by human judges and is therefore a useful measure which an automatic evaluation system would aim at determining automatically. It is important to note that responsiveness was not meant to be an absolute grading measure, but rather a partial ordering of the summarization abilities of the peers [Dang 2005]. An automatic measure that could provide a similar ordering should strongly correlate to the responsiveness grade-ordering assigned by humans.

Since there appears to be no absolute measure of quality for a summary even for human judges, an automatic measurement would require at least one *model summary* (i.e., human-extracted summary produced as a reference for measuring the quality of the summaries produced by others), also called gold standard or reference summary. Then, it would be possible to judge the *peer summaries* (i.e., summaries extracted by peer systems). Such measurements actually determine the distance between the peer and the model summaries. The questions posed for such an automatic measure with the same utility as the one responsiveness provides are the following.

¹Also see <http://duc.nist.gov/>.

- What is the kind of information that can be used in order to represent the peer and model summary?
- What should the actual representation method for the extracted information be in order to retain information valuable in the comparison process?
- What kind of similarity measure can be used or defined in order to provide meaningful results?

Automatic methods for the evaluation of summaries exist [Hovy et al. 2005a; Lin 2004; Zhou et al. 2006] and have a high correlation to the measure of responsiveness. There are, however, some desired characteristics that do not coexist in a single method.

- Language-neutrality. A method that does not require language-dependent resources (thesauri, lexica, etc.) and can be applied directly to different languages.
- Full automation. A method that does not require human intervention apart from the human model summaries.
- Context-sensitivity. A method should take into account contextual information so that the *well-formedness* of text is taken into account. Well-formedness can be loosely defined as the quality of a text that allows for easy reading. A text that is a random sequence of words would lack this quality even if the words are on a specific topic.

Our method, named AutoSummENG (AUTOMATIC SUMMERY Evaluation based on N-gram Graphs), attempts to have all these qualities, while bearing results with a high correlation to the responsiveness measure, which indicates a correlation to human judgement. The results of our experiments indicated that our method outperforms current state-of-the-art systems in this sort of correlation, while remaining strictly statistical, automated, and context-sensitive due to the nature of the representation used, namely, the n-gram graph (more on this in Section 3).

1.2 Structure of the Document

Within this document, we present an overview of the AutoSummENG method, followed by a brief introduction to required background knowledge and related work. After that, we elaborate on our method, both in terms of the proposed representation of summaries as well as the proposed similarity measures that can be applied upon these representations. In Section 3.1, we support the selection of the presented representation via experimental results. Then, a second part concerning a detailed model for a priori estimation of the method's parameters is presented and evaluated. In addition, experiments indicative of the high evaluation performance of the proposed method are presented in the Section 5. We close the article with conclusions and future work.

2. SUMMARY OF METHOD AND BACKGROUND KNOWLEDGE

The AutoSummENG method is based on the concept of using statistically extracted textual information from summaries integrated into a rich representational equivalent of a text to measure similarity between generated summaries and a set of model summaries. The novelty of the method concerns the following points:

- the type of statistical information extracted,
- the representation chosen for the extracted information,
- the method of similarity calculation.

The information extracted from source texts is a set of indicators of neighborhood between n-grams contained within the source text. In other words, the method proposes the extraction of *relations* between n-grams given spatial proximity of those n-grams within a given text.

Then a graph is constructed to indicate the full set of relations deduced (as edges between n-gram-labeled vertices), together with additional information concerning these relations. Such representations are extracted from both generated and model (i.e., human-composed) summaries. The edges of the graphs may contain such information as the mean distance between the neighboring n-grams in all occurrences, or a distance-weighted co-occurrence count for any given pair of n-grams, or even a detailed distribution of distances between pairs of n-grams in texts.

Finally, a comparison between the graph representation of generated and model summaries is made, indicating the degree of similarity between the graphs. At this point, generated summaries that are found to be, on average, more similar to model summaries are considered better. Systems that generate, on average, better summaries are in turn considered better systems. The comparison methodology varies from vertex-only comparison between graphs to full comparison including the information attached to the edges.

Given this, we have evaluated different representation types based on both the type of represented data (character or word n-grams) as well as the use or lack of use of connectivity information between the data (graph or histogram). During its evaluation, the system was found to perform differently based on its parameters. Therefore, a full study was conducted focusing on how these parameters can be a priori optimized to provide a fully automated evaluation methodology. The study was based on the fact that there are relations between meaningful n-grams that we call symbols and nonmeaningful ones, which we call ‘non-symbols’. These categories of n-grams are based on statistical criteria and are used to describe how noise can deteriorate the performance of our method as a function of the methodology parameters. Given this noisy-channel model of our approach, we were able to perform an a priori estimation of the method parameters.

At this point, we will review underlying theory used in evaluation of summarization systems as well as existing methods of such evaluation. We begin by indicating how the performance of evaluation methods is measured along with the required statistical measures, then describe existing approaches for

the evaluation of summaries and summarization systems. Then we provide background related to basic concepts of our methodology, such as n-grams and graphs, also presenting how comparisons between graphs are performed and the use of graphs in the domain of automatic summarization.

2.1 Measuring Correlation—Evaluation Method Performance

In the automatic evaluation of summarization systems, we require automatic grades to correlate to human grades. The measurement of correlation between two variables provides an indication of whether two variables are independent or not. Highly correlated variables are dependent on each other often through a linear relationship. There are various types of correlation measures, called *correlation coefficients*, depending on the context in which they can be applied. Three types of correlation will be briefly presented here since they are related to the task at hand.

- The Pearson’s product moment correlation coefficient reflects the degree of linear relationship between two variables.² The value of Pearson’s product moment correlation coefficient ranges from -1 to 1 , where 1 indicates perfect positive correlation and -1 perfect negative correlation. Perfect positive correlation indicates that there is a linear relationship between the two variables and that when one of the variables increases, so does the other in a proportional manner. In the case of negative correlation, when one of the two variables increases, the other decreases. A value of zero in Pearson’s product moment correlation coefficient indicates that there is no obvious correlation between the values of two variables.
- The Spearman’s rank correlation coefficient [Spearman 1906] performs a correlation measurement over the ranks of values that have been ranked before the measurement. In other words, it calculates the Pearson’s product moment correlation of the ranking of the values of two variables. If two rankings are identical, then the Spearman’s rank correlation coefficient will amount to 1 . If they are the reverse of each other, then the correlation coefficient will be -1 . A value of zero in Spearman’s rank correlation coefficient indicates that there is no obvious correlation between the rankings of values of two variables. It is important to note that this coefficient type does not assume linear relation between the values since it uses rankings. However, it presumes that subsequent ranks indicate equal distance between the corresponding values of the measured variable.
- The Kendall’s tau correlation coefficient [Kendall 1962] places one more limitation on the previous methods: it does not expect subsequent ranks to indicate equal distance between the corresponding values of the measured variable.

The previous correlation coefficients have all been used as indicators of performance for summary systems evaluation [Lin 2004; Nenkova 2006]. To clarify

²The linear relationship of two correlated variables can be found by using methods such as linear regression.

how this happens, consider the case where an automatic evaluation method is applied on a set of summarization systems, providing a quantitative estimation of the latter's performance by means of a grade. Let us say that we have assigned a number of humans to the task of grading the performance of the same systems. If the grades appointed by the method correlate to the grades appointed by humans, then we consider the evaluation method good. Let us now elaborate on how summary evaluation has been described and conducted according to the literature.

2.2 Summary Evaluation

A summary, even though it is supposed to be a reduced version of original texts, is, in any case, a text by itself. This applies to the model summary as well as the peer summaries. Therefore, one needs a representation of text which will enable semantic similarity determination between two texts. This representation would be an intermediate one, that would ideally allow us to position the original textual information in a metric space. However, in previous publications, the presented measures do not necessarily abide by the triangle inequality and are rather simple similarity measures.

A number of different intermediate representations of summaries' information have been introduced in existing summarization evaluation literature, ranging from automatically extracted snippets to human-decided subsentential portions of text. More specifically, the family of BE/ROUGE³ [Hovy et al. 2005a; Lin 2004] evaluation frameworks uses statistical measures of similarity based on n-grams of words,⁴ although it supports different kinds of analysis, ranging from n-gram to semantic [Hovy et al. 2005b]. The intuition behind the BE/ROUGE family is that in order for two texts to have similar meaning, they must also share similar words or phrases. One can take into account simple unigrams (single words) in the similarity comparison, or require larger sets of words to be shared between compared texts. *Basic elements* (BE) are considered to be the head of a major syntactic constituent and its relation to a single dependent. BEs are decided upon in many ways, including syntactic parsing and the use of cutting rules [Hovy et al. 2005b]. BEs can be matched by simple string matching or by more complex matching methods like semantic generalization and matching, according to a proposed framework [Dang 2005; Hovy et al. 2006].

A different approach [van Halteren and Teufel 2003] uses *factoids*, which are predicate-like structures representing summary information, while the approach in Passonneau et al. [2006] uses Semantic Content Units (SCUs) of the Pyramid annotation method. These representation types—factoids, SCUs—represent elementary information units that map semantic information items to different surface appearances.

Elaborating, the factoid concept specifies atomic pieces of information that can differentiate summaries. These factoids are of various lengths; they are

³See also Papineni et al. [2001] for the BLEU method on machine translation.

⁴We remind the reader that *N-grams of words* are groups of *words* with N elements. *N-grams of characters* are groups of *characters* with N elements.

represented in a manner similar to first-order predicate logic; and they are subjective. The main directives on which factoid annotation was based as indicated in van Halteren and Teufel [2003] was that a factoid could generally be represented by a simple first-order predicate logic predicate. Additionally, potential factoids always appearing together within a (source) text were viewed as one joined factoid, containing the union of all corresponding information. Systems with a high F-measure, that is, $\frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$, concerning factoids appearing in a gold-standard summary, were deemed better. On the other hand, SCUs are annotated semantically-adequate fragments of a text, no longer than a sentence, which are defined by humans based on a set of predefined rules. The notion of semantically adequate supposes that no useless fragments of information will be used as SCUs.

On the other hand, SCUs, need not be atomic or predicate-like. They are mere semantic equivalents of different surface appearances. In other words, they represent the common meaning of a set of phrases. SCUs carry a varying weight, which is calculated in terms of the number of appearances each SCU has within the model summaries. Thus, SCUs appearing in many model summaries are considered salient. The evaluated summaries were given a higher score if more high-weight SCUs appeared within them (see also Nenkova [2006] for more information).

Other human-decided information items appearing in the literature are *information nuggets* [Voorhees 2003], which were defined as facts for which the (human) “assessor could make a binary decision as to whether a response contained the nugget”. The information nuggets were used in the question answering track of TREC 2003, which explains what the word response refers to in the previous definition. Summaries containing more and distinct information nuggets were assigned a higher score.

The use of factoids, SCUs, information nuggets, and similar information representations, require human assessment, which can be costly. Therefore, even though the definition of frameworks that reduce interjudge disagreement are a necessity to perform more robust evaluation, one needs to devise automated evaluation methods to avoid the cost of human intervention.

The family of BE/ROUGE evaluation frameworks support such automated evaluation. They rely on the statistical analysis of co-occurring word n -grams between the peer and reference summary. There exist different models of n -gram analysis methodology, for example, ROUGE-2 identifies common bigrams, while ROUGE-SU4 is a method where bigrams with a maximum gap of four words between them are considered matching bigrams between the evaluated and model summary (also consult Lin [2004] for more information). Although ROUGE takes into account contextual information, it remains at the word-level, which means we either regard different types of the same word as different or we need to apply (language-dependent) stemming or lemmatization to remove this effect.

Basic elements (BE), on the other hand, use extraction techniques based on the use of structured representation in the form *head-modifier-relation*, where head is a major syntactic constituent and relation is the relation holding between modifier and head. The BE approach can use several preprocessing

modules to extract head terms, according to different parsing and cutting rules [Hovy et al. 2005b], which are language-dependent. Therefore, this method cannot be easily applied when the required linguistic modules do not exist for the specified language of the summaries. Thus, one would prefer a method to be language-neutral to avoid prerequisites.

The approach of ParaEval described in Zhou et al. [2006], uses a method based on machine translation practices where a paraphrase table is created based on parallel aligned corpora in two different languages. The assumption made is that “if two English phrases are translated into the same foreign phrase with high probability (shown in the alignment results from a statistically trained alignment algorithm), then the two English phrases are paraphrases of each other” [Zhou et al. 2006]. This kind of table is extracted automatically even though the parallel corpus should already exist. Then a three-level comparison between the peer and the model summary is conducted where, in the first-level, paraphrases are located in the two summaries. If, for some portions of texts, this matching process fails, the evaluation attempts to locate synonym words. Finally, if the latter match fails, simple lexical matching occurs.

The results of the application of Pyramid evaluation, and ROUGE-2, ROUGE-SU4 on the DUC 2005 data, indicated high correlation to the responsiveness measure, and thus the measures were considered useful. The method presented herein matches and even exceeds the correlation of the aforementioned methodologies on the newer, DUC 2006⁵ data in a language-neutral, statistical manner, while taking into account contextual information.

2.3 N-Grams and Q-Grams

In various applications of information extraction and retrieval as well as natural language processing, there have been a number of uses for the *n-gram* aspect of texts. An *n-gram* is a, possibly ordered, set of words or characters, containing *n* elements. Character *n-grams* are also called *q-grams* [Ukkonen 1992], but the term *q-grams* is used mainly in database indexing and string matching research, thus we will refer to *q-grams* as *character n-grams* to facilitate all readers. *N-grams* have provided a useful representation for tasks like text-categorization [Cavnar and Trenkle 1994], machine translation [Zens and Ney 2006], authorship identification [Houvardas and Stamatatos 2006], and summarization and summary evaluation [Banko and Vanderwende 2004; Lin and Hovy 2003; Copeck and Szpakowicz 2004].

Especially in the automatic summarization domain, *n-grams* appear as word *n-grams*, either used in the evaluation or the summarization process itself (e.g., in the ROUGE/BE family of evaluator methods [Hovy et al. 2005b, Lin 2004]). *N-gram* fuzzy matching detects similar portions of text even if other words appear between the *n-gram* words in the text [Lin 2004]. There are different methods to allow for varying degrees of flexibility in the comparison, indicating a trade-off between precision and laxity.

⁵Also see <http://duc.nist.gov/>.

2.4 Graph-Based Methods and Graph Matching

For some time, graphs have been apparent in artificial intelligence as a means of representing related objects. In the domain of automatic summarization, graphs have been used as a means to determine salient parts of text [Mihalcea 2004; Erkan and Radev 2004a, 2004b] or determine query-related sentences (see Otterbacher et al. [2005] for a related sentence retrieval methodology). A text can be represented as a graph either to indicate simple, lexical relationships [Mohamed and Rajasekaran 2006], or more complex, rhetorical structure [Marcu 2000], and even nonapparent information [Lamkhede 2005].

In multidocument summarization, graphs have also been used to detect differences and similarities between source texts [Mani and Bloedorn 1997] and interdocument relations [Witte et al. 2006] as well as relations of various granularity from cross-word to cross-document as described in cross-document structure theory [Radev 2000]. We also find the application of graphs to be a useful representation for multidocument summarization, for example, in Mihalcea [2005] where different iterations over graph representations of texts determine the salient terms over a set of source texts. Saliency has also been determined by the use of graphs, based on the fact that documents can be represented as small world topology graphs [Matsuo et al. 2001].

Graphs have not been used widely in the summary evaluation domain even though they can represent relations of complex structure. Probably this is linked to the fact that graph comparison or graph matching is a nontrivial process.

There are a number of known methods for the calculation of graph similarity, classified in two main categories.

—*Isomorphism-Based*. Isomorphism is a bijective mapping between the vertex set of two graphs V_1, V_2 , such that all mapped vertices are equivalent, and every pair of vertices from V_1 shares the same state of neighborhood as their corresponding vertices of V_2 . In other words, in two isomorphic graphs, all the nodes of one graph have their unique equivalent in the other graph, and the graphs also have identical connections between equivalent nodes. Based on the isomorphism, a *common subgraph* can be defined between V_1, V_2 as a subgraph of V_1 having an isomorphic equivalent graph V_3 , which is a subgraph of V_2 as well. The *maximum common subgraph* of V_1 and V_2 is defined as the common subgraph with the maximum number of vertices. For more formal definitions and an excellent introduction to the error-tolerant graph matching, that is, fuzzy graph matching, see Bunke [1998].

Given the definition of the maximum common subgraph, a series of distance measures have been defined using various methods of calculation for the maximum common subgraph or similar constructs like the maximum common edge subgraph, or maximum common induced graph (also see Raymond et al. [2002]).

—*Edit-Distance Based*. Edit distance has been used in fuzzy string matching for some time now, using many variations (see Navarro [2001] for a survey on approximate string matching). The *edit distance* between two strings corresponds to the minimum number of edit character operations (namely

insertion, deletion, and replacement) needed to transform one string to the other. Based on this concept, a similar distance can be used for graphs [Bunke 1998]. Different edit operations can be given different weights to indicate that some edit operations indicate more important changes than others. The edit operations for graphs are node deletion, node insertion, node substitution, edge deletion, edge insertion, and edge substitution.

Given graph representations of text, these graph matching methods can be used as a means to indicate text similarity as will be shown in the description of our method.

3. PROPOSED EVALUATION METHOD

Tackling the problem of what kind of information should be used to represent a peer and a model summary in the evaluation of a summary, one should take into account that the surface appearance of two equivalent pieces of the same semantic content need not be identical as happens in the case of paraphrases [Zhou et al. 2006]. Nevertheless, it is quite probable that the words expressing the content will exist in the same context or that part of the words used will be identical, for example, if different inflections are used. For more on the linguistic aspect of this assumption, please consult Manning and Schütze [1999]. Our method accounts for these assumptions while retaining language-neutrality by using only statistical methods and language-independent assumptions for the extraction of information from texts and for the computation of textual similarity.

3.1 Representation

Trying to capture more than the simple cooccurrence of words and in order to allow for different types of the same word, our method uses character n-grams positioned within a context-indicative graph. We call this construct an *n-gram graph*. In our analysis, we consider that we view neighborhood with respect to a current n-gram, which is a subsequence of the text analysed. In the following analysis, we have also used word n-grams to be able to evaluate the method since the n-gram graph representation is applicable to both word or character n-grams.

We have tested three methods concerning how the n-gram graph can be constructed, based on how neighborhood between adjacent n-grams is calculated in a text. In general, a fixed-width window of characters (or words) around a given n-gram N_0 is used with all characters (or characters) within the window considered to be neighbors of N_0 . These neighbors are represented as connected vertices in the text graph. The edge connecting the neighbors is weighted, indicating, for example, the distance between the neighbors or the number of cooccurrences within the text. Based on different types of windows, we can use the following approaches.

—*The nonsymmetric approach.* A window of length n runs over the summary text. If N_0 is located (i.e., begins at) at position p_0 , then the window will span from $p_0 - 1$ to $p_0 - n$, taking into account only preceding characters

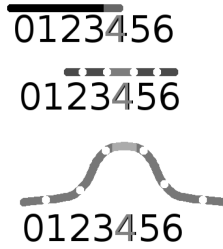


Fig. 1. Different types of n-gram windows (top to bottom): nonsymmetric, symmetric, and Gauss-normalized symmetric. N-gram 4 is the n-gram of interest.

or words. Each neighborhood-indicative edge is weighted by the number of cooccurrences of the neighbors within a given window of the text.

- The symmetric approach.* A window of length n runs over the summary text, centered at the beginning of N_0 . If the n-gram we are interested in is located at position p_0 , then the window will span from $p_0 - \lfloor \frac{n}{2} \rfloor$ to $p_0 + \lfloor \frac{n}{2} \rfloor$, taking into account both preceding and following characters or words. Each neighborhood-indicative edge is weighted based on the number of window cooccurrences of the neighbors, as previously indicated within the text.
- The Gauss-normalized symmetric approach.* A window of length $3 \times n$ runs over the summary text, centered on the beginning of our current n-gram N_0 . If N_0 is located at position p_0 , then the window will span from $p_0 - \lfloor \frac{3 \times n}{2} \rfloor$ to $p_0 + \lfloor \frac{3 \times n}{2} \rfloor$ (where $\lfloor x \rfloor$ gives the integer part of x), taking into account both preceding and following characters and words. However, in this case, the distance of a neighbor n-gram to the current n-gram is taken into account. In other words, an n-gram N_1 with distance d_1 from the beginning of N_0 , positioned at p_0 , is considered to be less of a neighbor than n-gram N_2 , positioned at distance d_2 , $d_2 < d_1$ from p_0 . Therefore, each neighborhood-indicative edge is weighted based on the number of cooccurrences of the neighbors within the text and the neighbors' distance at each occurrence. Also, the Gauss-normalized symmetric approach takes into account neighbors outside the given window size D_{win} to a full distance of $3 \times D_{\text{win}}$. This distance was selected given the fact that this accounts for 99.99% of the mass under the Gaussian distribution, given that we consider a standard deviation of D_{win} ⁶; that is to say, n-grams outside that distance have almost no effect. Thus, it is better in terms of complexity to just ignore those outliers.

In Figure 1, schematic representations of the three approaches have been illustrated. The numbers indicate adjacent n-grams which can either be word n-grams or character ones. The line over a number indicates that the n-gram has been taken into account as a neighbor. In the third part of the figure, the bell-shaped line indicates the different weights assigned to different distances from the n-gram positioned at p_0 . The latter n-gram, we call *n-gram of interest*,

⁶This can be easily derived by using the probability mass function of the Gaussian distribution: $\text{pdf}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$. See also Duda et al. [2001] and Appendix A.

Algorithm 1. Extraction of n-grams

```

Input: text
Output: n-gram set
// T is the text we analyse
1 for all  $i$  in  $[1, \text{length}(T)-n+1]$  do
2 | get substring of T from index  $i$  to  $i+n-1$ 
3 end

```

is indicated by the emphasized number in the figure. We found, through a set of experiments, that the most promising approach was the symmetric one, which may indicate that there is indeed a maximum distance outside of which relations do not hold.

We now provide the definition of n-gram, given a text (viewed as a character sequence).

Definition 3.1. If $n > 0, n \in \mathbb{Z}$, and c_i is the i -th character of an l -length character sequence $T^l = \{c_1, c_2, \dots, c_l\}$ (our text), then a *character n-gram* $S^n = (s_1, s_2, \dots, s_n)$ is a subsequence of length n of $T^l \iff \exists i \in [1, l - n + 1]: \forall j \in [1, n]: s_j = c_{i+j-1}$. We shall indicate the n-gram spanning from c_i to $c_k, k > i$, as $S_{i,k}$, while n-grams of length n will be indicated as S^n .

The meaning of the formal specification, is that n-gram S^n can be found as a substring of length n of the original text, spanning from the i -th to the j -th character of the original text. For example, if the text is the following sentence:

Do you like this summary?

then $S_{1,2}$ is the sequence $\{‘D’, ‘o’\} \equiv ‘Do’$ for display purposes.

Extracting N-Grams

If we choose to extract the n-grams (S^n) of a text T^l , the (elementary) algorithm is indicated as Algorithm 1.

The algorithm applies no preprocessing (such as extraction of blanks, punctuation, or lemmatization). Furthermore, it obviously extracts overlapping parts of text as the window is shifted by one position at a time and not by n positions at a time. This technique is used to avoid the problem of segmenting the text. The redundancy apparent in this approach proves to be useful similar to a convolution function: summing similarities over a scrolling window may prove more useful if you do not know the exact center of the match. In other words, the extracted n-grams are certain to result in a match to n-grams of the model summary if such an n-gram exists, whereas a method where the text was segmented in equal n-grams might not locate the matching n-grams.

Example 3.2. Application of our method to the sentence we have used previously with a requested n-gram size of 3 would return:
 $\{‘Do’, ‘o y’, ‘you’, ‘ou ’, ‘u l’, ‘li’, ‘lik’, ‘ike’, ‘ke ’, ‘e t’, ‘th’, ‘thi’, ‘his’, ‘is’, ‘s s’, ‘su’, ‘sum’, ‘umm’, ‘mma’, ‘mar’, ‘ary’, ‘ry?’\}$

while an algorithm taking disjoint n-grams would return {'Do ', 'you', ' li', 'ke ', 'thi', 's s', 'umm', 'ary'} (and '?' would probably be omitted).

In the case of the word n-gram extraction, the text is considered to be a word sequence (as opposed to a character sequence). The text has been preprocessed, using a simple tokenizer based on punctuation and blanks, to determine word boundaries. However, it is important that this simple tokenizer deprives the method of its complete language neutrality if used. Therefore, we will prefer the character n-gram version to the word n-gram version if their use in evaluation renders equivalent results.

Go back to the character n-gram version. It is rather obvious that segmentation should be done carefully in order to avoid redundancy without losing information on important sequences of characters. Consider the case where we match character n-grams between two segmented texts. In the given example, the fact that the word *summary* has been broken down into three disjoint n-grams may cause a mismatch or not match at all. For n-grams of higher length, or *rank* as it is called, the effect of information loss in the case of a careless segmentation may prove more deteriorating if we consider two n-grams to match if and only if they are exactly the same. Perhaps other methods of string comparison, like substring comparison, may decrease this loss. However, within this method, we will use simple string matching between n-grams.

The segmentation process by itself, even if one uses our approach, does not keep information concerning the relative position of n-grams in the original text; it only extracts n-grams. What this means is that we do not know if the n-gram *Do* is next to the n-gram *you*, or not. Thus, words (n-grams) that consist of what is called a *collocation*, that is, that when found together, they possess a meaning that is not simply a concatenation or composition of each separate word meaning [Manning and Schütze 1999], will lose their connection when extracted. This is where the graph representation comes in.

N-Gram Graph

The *n-gram graph* is a graph $G = \{V^G, E^G\}$ which has n-grams as its vertices $v^G \in V^G$, and the edges $e^G \in E^G$ (the superscript G will be omitted where easily assumed) connecting the n-grams indicate proximity of the corresponding vertex n-grams (see Figure 2). The edges can be weighted by applying the distance between the two neighboring n-grams in the original text. If the n-grams are found to be neighbors more than once in the original text, which is usually the case, one could assign a distribution of distances to each edge to use the distance information. The formal definition is as follows.

Definition 3.3. If $S = \{S_1, S_2, \dots\}$, $S_k \neq S_l$, for $k \neq l$, $k, l \in \mathbb{N}$ is the set of distinct n-grams extracted from a text T^l , and S_i is the i th extracted n-gram, then G is a graph where there is a bijection (one-to-one and onto) function $f : S \rightarrow V$.

The edges E are assigned weights of $c_{i,j}$, where $c_{i,j}$ is the number of times a given pair S_i, S_j of n-grams happen to be neighbors within some distance D_{win} (in characters for character n-grams) of each other or within a function

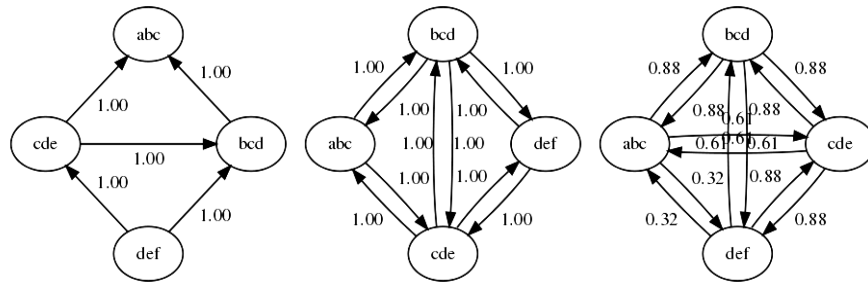


Fig. 2. Graphs extracted from the string *abcdef*, based on the three types of windows (left-to-right): nonsymmetric, symmetric and Gauss-normalized symmetric. The n-grams are character n-grams of rank 3.

of the given distance D_{win} as in the Gauss-normalized approach. Since probably not all distances are of importance, and thus two n-grams neighboring by 150 characters probably have no actual relation, we take into account only a window around S_i in the original text to determine which S_j deserves our attention. The vertices v_i, v_j corresponding to n-grams S_i, S_j that are located within this parameter distance D_{win} are connected by a corresponding edge $e \equiv \{v_i, v_j\}$.

Following this method of representation, we have reached a point where we have kept some information for a determined n-gram length n and parameter distance D_{win} . It is nontrivial, though, to choose a single $\{n, D_{\text{win}}\}$ pair that can be optimal for n-gram extraction, independent of the text. If one chooses a very low value for n , then the relation between different n-grams can be taken into account only by augmenting the D_{win} parameter. However, in the case of a high D_{win} value, the fact that we only take into consideration whether the n-grams are neighbors and not their actual distance may prove detrimental to the information we keep. In other words, if our D_{win} is 50, then a neighbor by 1 character will be considered equally close to one with a distance of 50 characters.

On the other hand, if n is too high, then the information we gather for each text will be extremely redundant and will definitely cause consumption of more resources as well as make the matching process more time-consuming. This happens because there are much more unique 10-grams than 2-grams in any selected (adequately long) text of a natural language like English or Greek. Furthermore, the number of vertices of the graph G_n for rank n of n-grams will increase exponentially to the rank n of n-grams⁷ (subscript n of G_n will be omitted when easily assumed).

In order to tackle these problems we take into consideration n-grams of various ranks with a rather small maximum distance between them determined to be valued around the average word length of the text language. The selection of different n-gram ranks allows for matching of various length n-grams without use of the distance metric, while the small D_{win} allows small variations of

⁷The grammar of each language does not allow all combinations of alphabet characters, and thus the possible 5-grams of a language with 26-letter alphabet are not 26^5 but somewhat lower. See also Manning and Schütze [1999, Sections 2.2.1–2.2.2].

distances and provides less accuracy loss concerning the distribution of distances. However, the selection of an optimal n-gram rank range $[r_{\min}, r_{\max}]$ proved to be an issue worth investigating, and so we did (see Section 4).

In the research conducted, it was important to see if a histogram offers equally good results with a graph in the process of the evaluation. If that stood, it would mean that the graph representation should not be used altogether.

N-Gram Histogram

The n-gram histogram representation is a simple frequency histogram measuring n-grams occurrences. In other words, it simply indicates the number of times an n-gram appears without any neighborhood information. This representation will be used as a baseline to indicate whether neighborhood information is indeed important in our domain of application.

3.2 Comparison

In order to compare two summaries T_1 and T_2 , we need to compare their representations. Given that the representation of a text T_i is a set of graphs \mathbb{G}_i , containing graphs of various ranks, we propose the following similarity measures between graphs G^i, G^j of the same supposed rank n .

—*Cooccurrence Similarity (CS)*, indicating how many of the edges contained in graph G^i are contained in graph G^j . We define $e \in E^G \equiv e \in G$. Thus cooccurrence is defined as:

$$CS(G^i, G^j) = \frac{\sum_{e \in G^i} \mu(e, G^j)}{\max(|G^i|, |G^j|)}. \quad (1)$$

μ is the membership function, which returns 1 if e belongs to G , else it returns 0. Also $|G|$ is the number of edges $e \in G$. The definition causes CS to be symmetric, that is,

$$CS(G^i, G^j) = CS(G^j, G^i). \quad (2)$$

Also, CS takes values in $[0, 1]$ with a value of 1 indicating a full match of the two graphs even though edge weights are not taken into account. On the other hand, a value of 0 means that no edge from one graph exists in the other. In this measure, each matching edge contributes by $\frac{1}{\max(|G^i|, |G^j|)}$ to the sum. The CS is a normalized derivative of common graph distance measures, based on the maximum common subgraph [Bunke 1998].

—*Value Similarity (VS)*, indicating how many of the edges contained in graph G^i are contained in graph G^j , considering also the weights of the matching edges. In this measure, each matching edge e having weight w_e^i in graph G^i contributes $\frac{VR(e)}{\max(|G^i|, |G^j|)}$ to the sum, while not matching edges do not contribute (consider that if an edge $e \notin G^i$, we define $w_e^i = 0$). The *ValueRatio (VR)* scaling factor is defined as:

$$VR(e) = \frac{\min(w_e^i, w_e^j)}{\max(w_e^i, w_e^j)}. \quad (3)$$

The equation indicates that the ValueRatio takes values in $[0, 1]$ and is symmetric. It is easy to see that this allows the VS metric to retain the symmetry inherited from the CS equation part. Thus, the full equation for VS is:

$$\text{VS}(G^i, G^j) = \frac{\sum_{e \in G^i} \left(\mu(e, G^j) \times \frac{\min(w_e^i, w_e^j)}{\max(w_e^i, w_e^j)} \right)}{\max(|G^i|, |G^j|)}. \quad (4)$$

VS is a metric converging to 1 for graphs that share both the edges and similar weights, which means that a value of $\text{VS} = 1$ indicates a perfect match between the compared graphs.

The analogous measure of CS from graphs to histograms, which we name CS_H , is actually based on a binary decision of whether an element h of histogram H_1 with a value of v_1 also exists in a histogram H_2 independent of the value v_2 of the same element in H_2 . Each coexisting element h contributes to CS_H a quantity of $\frac{1}{\max(|H_1|, |H_2|)}$ and, therefore, the equation for CS_H is

$$\text{CS}_H(H_1, H_2) = \sum_{h \in H_1} \frac{\mu(h, H_2)}{\max(|H_1|, |H_2|)}, \quad (5)$$

where $\mu(h, H)$ equals to 1 if $\exists h \in H : v(h, H) > 0$, otherwise it equals to 0, where v is the value corresponding to h in histogram H . Also, $|H|$ is the number of elements in histogram H .

On the same basis and by setting $v(h, H)$ of $h \notin H$ to 0, VS_H is defined as

$$\text{VS}_H(H_1, H_2) = \sum_{h \in H_1} \frac{\mu(h, H_2) \times v(h, H_2)}{\max(|H_1|, |H_2|)}. \quad (6)$$

The fact that we have proposed extraction of different ranks of n-grams for the composition of the graphs allows us to take advantage of matching between different ranks. The overall result of both the CS and the VS calculations, depicted by the superscript ‘O’, is a weighted average of the ranks. For example, for CS we have

$$\text{CS}^O = \frac{\sum_{r \in [L_{\min}, L_{\max}]} r \times \text{CS}^r}{\sum_{r \in [L_{\min}, L_{\max}]} r}, \quad (7)$$

where CS^r is the CS measure for extracted graphs of rank r , and L_{\min} , L_{\max} are arbitrary chosen minimum and maximum ranks.

The intuition behind Equation (7) is that matching in higher-order ranks is more important than matching in lower-order ranks. This intuition relies on the fact that languages rely on the composition of simpler character strings to create more complex ones which bear richer meaning than their components. This composition occurs in characters, words, sentences, paragraphs, and so forth and has been founded both by generative as well as statistical language processing research (e.g. Manning and Schütze [1999, Introduction]), similarly for VS^O , CS_H^O , VS_H^O . In the experiments, these overall values were used as results for the comparison process. It should be noted

that the function of weight given the rank has been found empirically, and thus better alternatives can be found.⁸

3.3 Experimental Setup—Representation

Based on the definition of the proposed similarity measures, we wish to show by experiments that systems with summaries getting higher CS or VS than others are indeed better systems, and so it has proved to be, by correlation to the responsiveness measure (see table VI). We will refer to this correlation to the responsiveness measure as *evaluation performance* (EP) as opposed to system or method performance which refers to the grade appointed to an evaluated system or method.

The data on which the measurements were conducted was the summary and evaluation corpus of DUC 2006. The corpus consists of summaries for 50 different topics. Each topic had a number of automatically extracted summaries, one for each participating system, and 4 human created summaries. The human summaries were differentiated by means of an identifier as were the baseline system summaries, which originated from a baseline system created by NIST and simply took the first 250 words of the most recent document for each topic. It is important to indicate that human summaries were used both as model summaries and as peer summaries. All summaries were truncated to 250 words before being evaluated. To verify some of our experiments using a second corpus, we have used the corpora of DUC 2005 and DUC 2007 as well. The corpus of DUC 2005, similar to the one of DUC 2006, consists of summaries for 50 different topics. The only major difference is that 30 of the topics had 4 human summaries each, while the remaining 20 topics each had either 9 or 10 human summaries. The corpus of DUC 2007 consists of summaries for 45 different topics. Each topic has 4 humanly created summaries as well as 28 machine generated summaries.

The measure used for evaluation performance was the correlation of the method evaluation metric to the responsiveness measure, which has already been used and accepted by recent research [Dang 2005; Hovy et al. 2005; Nenkova 2006]. The statistical measures used were Pearson’s product moment correlation coefficient as well as Spearman’s rank correlation. We remind the reader that Pearson correlation takes values in $[-1, 1]$ where a value of 1 (or -1) indicates perfect (or negative perfect) correlation between two measures, while a value of 0 indicates no apparent correlation. Spearman correlation indicates whether two measures used as grades will provide similar rankings given a set of contesting entities with values near 1 indicative of a higher correlation. On that basis, we would require our method to approach the maximum correlation value of 1 to the responsiveness measure. In the following tests, we consider representations and measures with Spearman’s coefficient values closer to 1 better. It should be noted that the experiments used both character and word n-grams as granularity levels to see if there is a difference between the two.

⁸We have also tried the simple reverse of rank $r(\frac{1}{r})$ with worse results.

The set of experiments concerning representation attempted to evaluate simple n-gram histogram representation in comparison to the graph representation. The measures of cooccurrence similarity (CS) and value similarity (VS) have been used for both representation schemes, and the results of the experiments were correlated to the responsiveness measure.

Each peer summary was compared both in terms of character n-gram and word n-gram graphs as well as the corresponding character and word n-gram histograms to all model summaries separately. Then the similarities were averaged to conclude a final similarity of the peer summary to the models. Human summaries that appeared both as peer and model summaries were not compared to themselves in the process of comparison.

Requiring that representations are compared between themselves, regardless of the parameters, we conducted a series of experiments within a very wide range of parameter settings. The three parameters used are:

- (1) minimum n-gram length, indicated as L_{\min} ,
- (2) maximum n-gram length, indicated as L_{\max} ,
- (3) neighborhood Window Size, indicated as D_{win} .

The values given to the parameters were as follows.

- $L_{\min} \in [1, 10]$, which means we have taken into account n-grams from uni-grams to ten-grams.
- $L_{\max} \in [L_{\min}, 10]$, which means we have taken into account n-grams from the size of the selected L_{\min} and up to ten-grams.
- $D_{\text{win}} \in [1, 20]$, which means we have taken into account a window size of one and up to twenty in different iterations of the experiment.

The features that differentiate representations are:

- the word or character nature of the n-grams in the representation,
- the type of the representation: histogram or graph,
- the existence of binary or real values in the representation (cooccurrence vs. value). For example, a histogram indicative of occurrence would only use binary values, while a histogram containing frequency or probability of occurrence would use real values. In the case of a graph, a binary value may indicate cooccurrence, while a real value may also indicate strength of this cooccurrence.

3.4 Results on Characters: Histogram or Graph—Cooccurrence or Value

Table I depicts the evaluation performance of four different approaches concerning the representation (either graph or histogram) and measure (either cooccurrence or value). For each approach, we have measured the maximum evaluation performance, the minimum evaluation performance, the mean and the standard deviation of evaluation performance. Concerning character n-grams, in Table I, we can see that the most promising representation is that of the graph value, based on the ranking of average performance and robustness (i.e., least standard deviation).

Table I. Histogram and Graph Character N-Gram Statistics Ranked by Mean Performance

| Representation-Measure | Max | Min | Mean | Std. Deviation |
|------------------------|-------|-------|-------|----------------|
| Graph-Value | 0.938 | 0.803 | 0.914 | 0.019 |
| Histogram-Cooccurrence | 0.934 | 0.723 | 0.912 | 0.035 |
| Graph-Cooccurrence | 0.912 | 0.738 | 0.883 | 0.020 |
| Histogram-Value | 0.854 | 0.502 | 0.633 | 0.097 |

Statistical Evidence

In order to statistically support whether different approaches indeed rendered different results and, thus, conclude which approach is better, we tested whether the distribution of evaluation performance values for each method was normal (Gaussian). If this stood, we would be able to apply a simple one-sided t-test to reach a decision about comparison between approaches. The Anderson-Darling normality test [Stephens 1974] was used to judge normality. This test has the null hypothesis that the samples examined come from a normal distribution. Unfortunately, the distributions of the samples used are not normal (the samples had a probability of less than 2.2×10^{-16} to originate from a normal distribution), and we had to use another test.

At first, we tested whether the distributions of results from all methods were all different from one another, which would indicate that there is no reason to choose a specific distribution. To do this, the Kolmogorov-Smirnov goodness-of-fit test [Massey Jr 1951] which has the null hypothesis that two sets of samples come from the same distribution, was used. The test indicated that all distributions are indeed different with a confidence of more than 99%.⁹

The test selected as appropriate for the task of deciding which evaluation approach performs better was the Wilcoxon rank sum test (also called Mann-Whitney test) [Hollander and Wolfe 1973, 27–33, 68–75]. This test ranks the samples (in our case the evaluation performance values) over a common axis (in our case the \mathbb{R} axis). Then, it uses the sum of the ranks of each sample set to see which corresponding distribution is more probable to give samples that stand higher than the samples of the other in the given axis. According to this test, the ranking of methods indicated that Histogram-Cooccurrence is probably a little better than Graph-Value methods, contrary to what the mean and standard deviation have indicated. In the bottom half of the ranking, Graph-Cooccurrence is worse than Graph-Value but better than Histogram-Value, as expected (See Table II).

There was a hint that the oddity concerning Histogram-Cooccurrence and Graph-Value may be due to the effect of the D_{win} parameter in Graph-Value results. In other words, there are a lot of samples from the Graph-Value approach that have extreme D_{win} values, which affect the whole distribution of results. To check this possibility, we performed the same test between Histogram-Cooccurrence and Graph-Value evaluation performances where we had kept

⁹The p-value of the null hypothesis was less than 2.2×10^{-16} for all cases except for the G-V—G-C case, where p-value was 0.001015, well over the 99% confidence level.

Table II. Statistics Using Spearman Correlation for Character N-Grams on DUC 2006. Each Square Indicates Whether the Representation of the Corresponding Row Gives Better Results than the One in the Corresponding Column, Based on a Wilcoxon Rank Sum Test with 95% Confidence Level. G Stands for Graph, H for Histogram, C for Cooccurrence and V for Value

| | <i>G - C</i> | <i>G - V</i> | <i>H - C</i> | <i>H - V</i> |
|--------------|--------------|--------------|--------------|--------------|
| <i>G - C</i> | – | No | No | Yes |
| <i>G - V</i> | Yes | – | No | Yes |
| <i>H - C</i> | Yes | Yes | – | Yes |
| <i>H - V</i> | No | No | No | – |

only performances, of the Graph-Value approach for $D_{\text{win}} \leq 10$, which is still a high value considering the average size of words in English. This indeed turned things around, and Histogram-Cooccurrence was presented as worse than Graph-Value even though the confidence level of 95% could not be reached (p-value of 0.1123). We double checked our intuition in the DUC 2005 corpus, once more for values of D_{win} under 10, inclusive, and got a result indicating that Graph-Value is better than Histogram-Cooccurrence with a confidence level of 99%. These indications gave us the incentive to delve further into the optimization of parameters (Section 4). However, all the analysis indicated that, given nonextreme values of all three parameters ($D_{\text{win}}, L_{\text{min}}, L_{\text{MAX}}$), the Graph-Value approach can outperform the Histogram-Cooccurrence one.

One should note that the D_{win} parameter does not affect the histogram representation (as there is no D_{win} parameter when using the histogram). This means that the test concerning the histogram only included L_{min} , and L_{MAX} parameters. The experiments concerning the graph representation, on the other hand, included D_{win} . We tested the effect of all parameters on the system performance.

To determine the effect of different parameters on the performance of the Graph-Histogram evaluation, we used a scatterplot graph. The scatterplot graph indicates how the values of our evaluation performance (vertical axis) as a value of correlation to human grading varied between different runs given different values of a parameter. Large variation in performance for a single parameter value is indicated by highly dispersed points in the vertical axis, while robustness is indicated by many, closely positioned points in the vertical axis. The smooth line in the graphs was extracted via LOWESS regression [Cleveland 1981] and helps identify the trend of the performance given the parameter.

In Figure 3, we can see that marginal values of L_{min} , which is the parameter under study, worsen the performance of the system. In low values, the deviation is raised, while, in high values, the average performance is lowered.

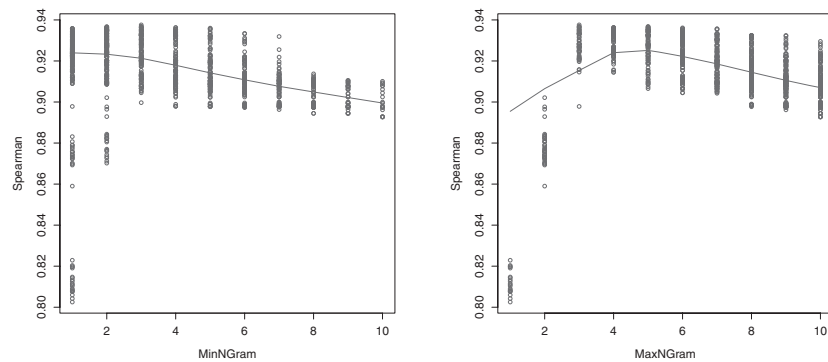


Fig. 3. Scatterplot of L_{\min} (MinNGram) on the left and L_{\max} (MaxNGram) on the right and Performance (Spearman)—Char N-Grams.

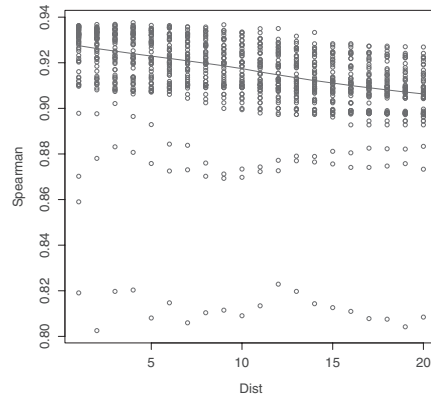


Fig. 4. Scatterplot of D_{win} (Dist) and Performance (Spearman)—Char N-Grams.

In Figure 3 we can see that, like L_{\min} , marginal values of L_{\max} worsen the performance of the system. In this case, there is no obvious effect in the robustness of the performance.

Finally, in Figure 4 we can see that, while there is no obvious effect in the robustness of the performance by increasing the value of the D_{win} parameter (depicted as Dist in the figure itself), the performance appears to deteriorate gradually.

In order to validate this impression, we examined the Pearson correlation between the three parameters and the performance of the system. The results, shown in Table III with the first column indicating the parameter and the second the correlation of its value to the evaluation performance, indicate that there is indeed negative correlation between D_{win} and performance, even though it is not very strong. The assumption derived from the negative impact of the distance to the performance was that we should be very careful with the selection of D_{win} , as it seems to insert some type of noise into the evaluation process.

Table III. Pearson Correlation between D_{win} and Performance for Character N-Grams Graph-Value

| Parameter | Correlation |
|-----------|-------------|
| D_{win} | -0.316 |
| L_{min} | -0.148 |
| L_{MAX} | 0.080 |

Table IV. Histogram and Graph Word N-gram Statistics Ranked by Mean Performance

| Representation-Measure | Max | Min | Mean | Std. Deviation |
|------------------------|-------|-------|-------|----------------|
| Histogram-Value | 0.898 | 0.450 | 0.767 | 0.108 |
| Histogram-Cooccurrence | 0.909 | 0.306 | 0.741 | 0.173 |
| Graph-Value | 0.909 | 0.072 | 0.478 | 0.227 |
| Graph-Cooccurrence | 0.882 | 0.046 | 0.457 | 0.223 |

3.5 Results on Words: Histogram or Graph—Cooccurrence or Value

Applying the same pattern of analysis for the word n-grams we reach the following conclusions.

- In word n-grams, the histogram is by far better than the graph (see Table IV).
- The two best approaches, both of which concern histograms, do not have statistically supported difference in their performance. Therefore, the simplest of the two should be chosen (i.e., Histogram-Cooccurrence).
- There is very serious variance in the results for different parameter values. The standard deviation is actually an order of magnitude higher than that of the character n-grams. This has to do with the impact of parameter values on performance. The word-based system seems to be much more sensitive than its character counterpart to L_{min} (see Figure 5 and Table V). On the other hand, L_{MAX} seems not to affect the histogram performance (see Table V). This indicates that for word n-grams, one should be very careful in choosing minimal values for the graph approach.

To recapitulate, our experiments indicate that, in the task of summary system evaluation:

- the best representation for character n-grams is the Graph—Value representation, even though the Histogram—Cooccurrence representation is almost equally effective when the distance parameter for the Graph—Value representation is extreme. A low value in the distance parameter of the graph is more likely to produce good results. The L_{MAX} parameter should be chosen to have a nonextreme value even though further experiments presented in Section 4 were conducted to show which value can be considered as nonextreme.
- the best representation word n-grams is that of Histogram—Value even though Histogram—Cooccurrence is not much less effective. In word n-grams, the minimum n-gram rank parameter L_{min} of the histogram plays a serious

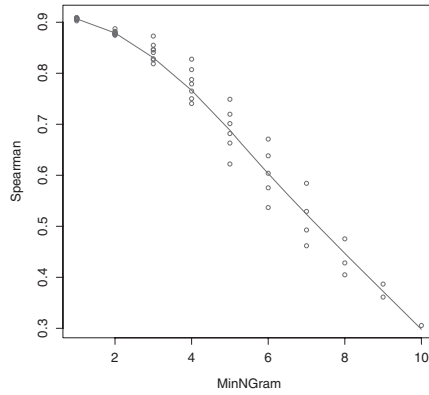


Fig. 5. Scatterplot of L_{\min} and Performance (Spearman)—Word N-Grams.

Table V. Pearson Correlation between D_{win} and Performance for Word N-Grams Histogram–Value

| Parameter | Correlation |
|------------------|-------------|
| L_{\min} | −0.255 |
| L_{MAX} | −0.038 |

role, indicating that low-rank n-grams are important and should be used, while the upper limit to the choice of n-gram rank is not directly linked to the overall performance and should therefore be kept low to reduce the number of calculations.

—considering the fact that the use of character n-grams performs overall much higher than its word counterpart (look again at Tables I and IV), we should examine the use of character n-grams further. As we discuss in Section 6, the word n-gram methodology should be examined under another point of view to see whether it correlates to other evaluation measures outside the scope of this work.

4. OPTIMIZING N-GRAM GRAPH PARAMETERS

Despite the robustness of the proposed method, we attempted to delve further into the strengths and weaknesses of the n-gram graph representation and the parameters inherent in the described method. We remind the reader that the three parameters used are:

- (1) minimum n-gram length, indicated as L_{\min} ,
- (2) maximum n-gram length, indicated as L_{MAX} ,
- (3) neighborhood Window Size, indicated as D_{win} .

The absolute limits of these three parameters are actually text-driven since all parameters cannot exceed the size of the text. There is an additional obvious restriction, demanding that the maximum n-gram length should not be lower than the minimum n-gram length. However, since the complexity of the data

structure and the number of calculations is exponential to the n-gram lengths as well the window size, we created a model that can predict near-optimal values for the parameters.

In order to verify the correctness of the model, as well as the deviation of its response from the actual optimal, we conducted a series of experiments in the corpus of DUC 2006 using two approaches:

- (1) the exhaustive approach, where a big number of combinations of the triplet L_{\min} , L_{\max} , D_{win} were evaluated within adequately big limits for each parameter to extract an overall optimum, and
- (2) the model-based approach, where the model-predicted values of the parameters were evaluated to indicate whether the response was approximate to the optimum.

During the exhaustive approach the parameters were given values as follows.

- $L_{\min} \in [1, 10]$, which means we have taken into account n-grams from uni-grams to ten-grams.
- $L_{\max} \in [L_{\min}, 10]$, which means we have taken into account n-grams from the size of the selected L_{\min} and up to ten-grams.
- $D_{\text{win}} \in [1, 20]$, which means we have taken into account a window size of one and up to twenty in different iterations of the experiment.

The limits of the given values were set arbitrarily, however, it was obvious during the experiments that performance of the system near the limits was very low, deteriorating with every step with higher parameter values. Thus, it was obvious that our limits were rational, given the language¹⁰ and the set of texts. It should be reiterated that we used the symmetric approach for the extraction of n-gram neighbors (see Section 3.1) in all our experiments because it exhibited the most promising results and is probably the most language-neutral, considering orientation in the writing of texts (left-to-right or right-to-left).

In order to reduce the number of experiments, at first we attempted to hold the D_{win} parameter constant in the arbitrary but well-performing value of 3 and change only the values of L_{\min} and L_{\max} . This way we planned to find a local-maximum and to investigate a correlation between n-gram size limits and system performance. In the course of our experiments, we discovered that the optimal value of D_{win} is correlated to L_{\min} , L_{\max} and cannot be held constant. At that point, we formed a model that would contain all our findings.

4.1 Symbols, Non-Symbols

We considered our text T to contain *symbols* and *non-symbols*. Let us elaborate on these two types of character sequences.

Symbols. They are supposed to carry the meaning of the text, and they should be sequences of characters (letters) that are not neighbors by mere chance. The

¹⁰A language where the average word length is twenty characters might require different limits.

| |
|---|
| 'permanent', 'permit', 'permits', 'persist', 'person', 'personal', 'personal computers', 'personnel', 'persons', 'persuade', 'pesticide', 'pesticides.', 'permi', 'permitt', 'pers', 'pers and', 'person kn', 'person or', 'perti', 'perty', 'pes', 'pes o' |
|---|

Fig. 6. Sample extracted symbols from the symbol process.

| |
|---|
| 'permit </HEADLINE>', 'permit program', 'permit approved' |
|---|

Fig. 7. Sample non-symbols.

letters of an existent word should be found neighboring more often than random characters that do not form a word.

Non-symbols. They are sequences of characters (letters) that simply happen to occur near each other and have no actual meaning by themselves. Non-symbols are all the letter sequences from the text that are not symbols.

In Figure 6, we have indicated some sample extracted symbols and in Figure 7 some non-symbols. We see that symbols may include simple terms and collocations (more on collocations in Manning and Schütze [1999, Sec. 1.4.4]). We can also see other sequences of letters, like words lacking their endings (e.g., permitt), ending themselves (e.g., perty, pes) or other sequences of no apparent semantics (e.g., pes o).

Ideally, we would like our representation to only include symbols. However, based on the given method, the n-gram graph includes information for both symbols and non-symbols, which induces noise in the comparison process. Therefore, we need to identify such values of parameters $L_{\min,0}$, $L_{\max,0}$, $D_{\text{win},0}$ that minimize the noise, while maximizing the quantity of useful information.

The following problems arise.

- How can we formally define and detect symbols and non-symbols?
- Is the usefulness of all symbols equal and how do we measure it?
- Can we define a single, easily understood quantity that we need to maximize in order to achieve the required result?

4.2 Detection of Symbols

As described previously, a symbol is a sequence of characters in the case of character n-grams. This sequence abides by a single rule: each letter is more probable to tail its preceding subsequence of characters than a character drawn randomly from a pool of characters. We elaborate on how a symbol is extracted from a text.

We have a text T^l . We denote by s_t the symbol we have composed at step t of the extraction process and c_t the candidate symbol for the t th step. A candidate symbol will become a symbol if and only if it conforms to the rule described in the previous paragraph. In order to determine the probability of a given substring X to be followed by a given character y , we construct a corpus containing a set of texts from the domain. In our case, we simply used all the model summaries

as the corpus, and we created an overall text, T_0^L of length L , formed by the concatenation of the corpus texts. Thus, given a pair (X, y) with X having a length of $|X|$, we can count:

- how many times X appears in T_0 , represented by N_X ,
- how many times the string Xy appears in T_0 , represented by N_{Xy} ,
- the total number of n-grams of a given size n within T_0 , represented by $|T_{0,n}|$.

We need to calculate the probability $P(y|X)$ of a given suffix y , given the prefix X :

$$P(y|X) = P(X) * P(y, X), \text{ where } P(y, X) = \frac{N_{Xy}}{|T_{0,n}|} \text{ and } P(X) = \frac{N_X}{|T_{0,|X|}|}. \quad (8)$$

On the other hand, the probability $P(y_r|X)$ of a random suffix y_r , given the prefix X , is given by:

$$P(y_r|X) = P(y_r),$$

since y_r is chosen randomly and independently from X .

Thus,

$$P(y_r) = \frac{1}{|A|_1}. \quad (9)$$

$|A|_n$ is the number of strings of length n , found in the alphabet A of T_0 . When we use the term *alphabet of T_0* , we refer to the set of unique characters appearing in T_0 . We select this definition of an alphabet because we want our approach to remain language-neutral, and therefore we do not presume any given alphabet.

The extraction of symbols from T_0^L is described as Algorithm 2.

Descriptively, the Algorithm 2 runs through the text, splitting symbols when the next character seems to have been positioned after the current substring by mere chance. Starting with a single-character candidate symbol, the algorithm adds new characters to the candidate symbol until a split point is reached. Then the candidate symbol is upgraded to a symbol, and a new candidate symbol is formed using the next character. In Figure 8, we can see the distribution of symbol sizes as extracted by the algorithm from the DUC 2006 corpus.

The aware reader may note that this method is related to the symmetric conditional probability used in the LocalMaxs algorithm [da Silva et al. 1999] as well as the notion of glue in Houvardas and Stamatatos [2006]. The main difference is that we do not evaluate candidate n-grams to keep the most prominent ones, but we consider all n-grams that represent symbols to be important and all others not important. Additionally, the probabilities used in the extraction of symbols here are different from the ones already in previous publications (e.g., SCP), and there are no fixed rules based on the n-gram size because these would be language-dependent. However, it would be interesting to see whether the use of other existing methods for variable rank n-gram extraction can prove more fruitful than the proposed one. This has not been done in the context of the current work due to the reasons stated previously.

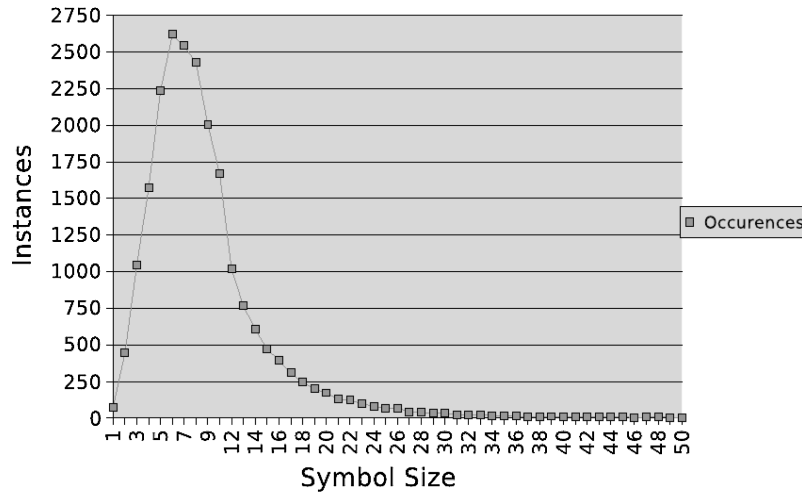


Fig. 8. The Distribution of symbols per rank (symbol size) in the DUC 2006 corpus.

Algorithm 2. Extracting Symbols

Input: text T_0^L
Output: symbol set S

// t denotes the current iteration, but has no use in the algorithm
// $T[i]$ denotes the i -th character of T
// ϵ is the empty string
// The plus sign (+) indicates concatenation where character series are concerned.

```

1  $S = \emptyset$ ;
2  $s_t = T[1]$ ;
3 for all  $i$  in  $[2, \text{length}(T)]$  do
4    $y = T[i]$ ;
5    $c_t = s_t + y$ ;
6   if  $P(y|s_t) > P(y_r)$  then
7      $s_t = c_t$ ;
8   end
9   else
10     $S = S + s_t$ ;
11     $s_t = y$ ;
12  end
13 end
    // Add last symbol
14  $S = S + s_t$ ;

```

4.3 Signal to Noise—Symbol Importance

In order to determine a measure of the importance of each symbol in our method, we insisted on the probabilistic approach. We consider any given symbol to be

more important if it is less probable that it is generated by a random symbol creator. This symbol creator, in order to create a new n -gram of size n , would choose randomly on $n - 1$ rank n -gram from a pool of valid symbols and would randomly select a 1-gram symbol to append, creating the new n -gram. The importance of a symbol is indicated by a weighting factor. On the other hand, we consider non-symbols to be equally (un)important in that each non-symbol has an importance of 1.

The fact that we have interesting and uninteresting pieces of data that form our input is analogous to a noisy channel model where a signal (interesting pieces) is transmitted over a medium (algorithm) that adds noise (uninteresting pieces). In this case, we would like to change the medium parameters (L_{\min} , L_{\max} , D_{win}) in order to maximize the signal and minimize the noise. A signal-to-noise approach, trying to predict the trade-off between different values of L_{\min} and L_{\max} concerning the signal-to-noise ratio, can be based on an equation like

$$SN(L_{\min}, L_{\max}) = 10 \times \log_{10} \left(\frac{S(L_{\min}, L_{\max})}{N(L_{\min}, L_{\max})} \right), \quad (10)$$

where $S(L_{\min}, L_{\max})$, $N(L_{\min}, L_{\max})$ are functions returning a measure of signal and noise correspondingly for a given range (L_{\min}, L_{\max}). SN indicates the function of signal-to-noise. The signal is the useful information we have captured via symbols, while the noise is the redundant or useless information we have captured via non-symbols.

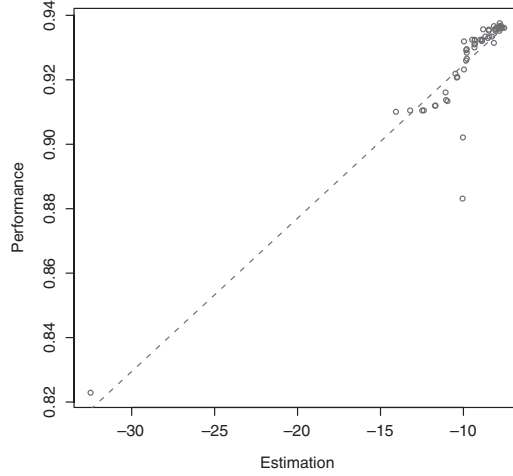
$N(L_{\min}, L_{\max})$ is defined as the count of non-symbols appearing in a given corpus for the given range:

$$N(L_{\min}, L_{\max}) = \sum_{i=L_{\min}}^{L_{\max}} |\text{Non-Symbols}_i|, \quad (11)$$

where $|\text{Non-Symbols}_i|$ is the number of symbols of rank i .

On the other hand, for the case of symbols, we wish to take into account the importance of each symbol, and therefore calculate *normalized weighted symbols*. The latter are weighted according to their importance, which is a function of their rank. The normalization step occurs over the weighted symbols to provide a new set of symbols, same in number as the ones found in the texts, which are, however, rearranged over different ranks in a way that they also illustrate the importance of any given rank. The number of weighted symbols for each n -gram rank r is calculated in two steps, within the given range $[L_{\min}, L_{\max}]$.

- (1) Calculate the weight w_r of symbols for the specified rank r and sum over all weighted symbols to find the total, weighted symbol sum W_r for rank r . The weight w_s is defined to be the inverse of the probability of producing a symbol of rank r given a symbol of rank $r - 1$ since longer symbols are less probable to appear as a result of a random sampling of characters.

Fig. 9. Correlation between estimation (SN) and performance .

Thus,

$$P(s_r | s_{r-1}) = \begin{cases} \frac{1}{|\text{Symbols}_r| + |\text{Non-Symbols}_r|} & \text{if } r = 1. \\ \frac{1}{|\text{Symbols}_{r-1}| + |\text{Non-Symbols}_{r-1}|} \times \frac{1}{|\text{Symbols}_r| + |\text{Non-Symbols}_r|} & \text{else.} \end{cases}$$

So $w_r = 1/P(s_r | s_{r-1})$, (12)

where $|\text{Symbols}_r|$ is the number of symbols in rank r .

- (2) Normalize W_r so that the sum of W_r over $r \in L_{\min}, L_{\text{MAX}}$ is equal to the original number of symbols in the texts. The normalized, weighted symbols W_r^0 for rank r are calculated by:

$$W_r^0 = W_r \times \frac{|\text{Symbols}_r|}{\sum_{i=L_{\min}}^{L_{\text{MAX}}} |\text{Symbols}_i|}. \quad (13)$$

We indicate once more that the W_r^0 measure actually represents the importance of symbols per rank r for the symbols of the texts, instead of the number of symbols per rank that is indicated by $|\text{Symbols}_r|$.

Thus, $S(L_{\min}, L_{\text{MAX}})$ finally equals to

$$S(L_{\min}, L_{\text{MAX}}) = \sum_{i=L_{\min}}^{L_{\text{MAX}}} W_i^0. \quad (14)$$

Having defined our signal-to-noise function, we wish to maximize it and, therefore, we search the space of parameter values for optimum values. However, we have to investigate whether our estimate of signal-to-noise is correlated to the performance of the system because only then will SN be useful. Indeed, SN offers an important 0.949 rank correlation (Spearman) to the maximum performance that can be achieved by our method (see Figure 9). The same correlation holds for the mean performance for a given L_{\min}, L_{MAX} pair and different values of D_{win} . In fact, SN is also almost linear to the performance

of the system with a Pearson correlation of 0.918. Therefore, our estimator is rather good in finding optimal values for L_{\min} and L_{\max} .

However, we have not yet discussed the distance parameter. We have said that it has a rather serious impact on the performance of the system. Up to this point, the D_{win} parameter was presumed to be independent from L_{\min} and L_{\max} . However, further evolution and testing of our model indicated a possible connection between D_{win} and L_{\max} .

We want to know, in the same manner as before, what the signal-to-noise ratio is as a function of distance D_{win} . We shall refer to this ratio as SN_d . In order to determine SN_d , we can once more count the symbols and non-symbols expected to be found in a given distance D_{win} from our n-gram. Let us consider, without harming generality, the case where we are in the middle of our (infinitely long) text, and we have a visibility of D_{win} characters to the right and left. Our current n-gram is of rank s_0 . We are extracting n-grams of size r .

During our n-gram extraction, we extract $d_0 = 2 \times D_{\text{win}}$ n-grams (for the symmetric approach). Thus, there are d_0 candidate symbols. In order to calculate the probability of extracting N_s symbols from d_0 attempts, we can model the process of extraction with a binomial success probability calculated for N_s successes in d_0 attempts. The chance of success for the binomial for a given n-gram rank of r is given by

$$P_s = \frac{W_r^0}{W_r^0 + |\text{Non-Symbols}_r|}. \quad (15)$$

The chance that our current n-gram is a symbol P_s^0 is the probability calculated by

$$\frac{W_{s_0}^0}{W_{s_0}^0 + |\text{Non-Symbols}_{s_0}|}. \quad (16)$$

Presuming that only information about neighboring symbols is important, the signal function should take into account only the probability of having both a symbol current n-gram and a symbol neighbor n-gram. Even though the maximum number of nonoverlapping neighbor symbols we can find within d_0 is $\lfloor \frac{d_0}{r} \rfloor$, we will not use this limitation in our analysis. We do so because the analogy of symbols and non-symbols remains the same on average over all our corpus, and our estimator can count on this analogy to extract good results on average. To extract an estimate E_s^r of the number of symbols that can be extracted for a given rank r , we use the algorithm indicated as Algorithm 3.

From Algorithm 3, we get an estimated number of symbols. The rest of the d_0 extractions are non-symbols and account for $d_0 - E_s^r$ extractions. Therefore, SN_d can be calculated by

$$SN_d(L_{\min}, L_{\max}) = 10 \times \log_{10} \frac{S_d(L_{\min}, L_{\max}, D_{\text{win}})}{N_d(L_{\min}, L_{\max}, D_{\text{win}})}, \quad (17)$$

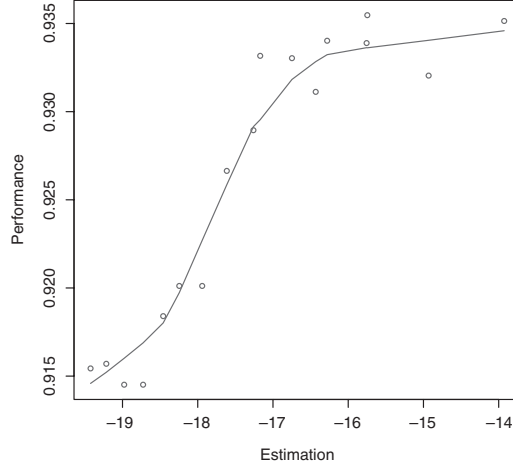


Fig. 10. Correlation between estimation (SN) and performance for given L_{\min} , L_{\max} .

Algorithm 3. Symbol Count Estimation

Input: Distance D_{win} , Success Probability of Single Trial P_s

Output: Expected Number of Symbols E_s^r

// $D(x)$ is a discrete probability distribution

1 for all i in $[1, D_{\text{win}}]$ **do**

2 | $D(x) = \text{binomial}(i; D_{\text{win}}, P_s)$;

3 end

// $E(y)$ is the mean function

4 $E_s^r = E(D(x))$;

where S_d , N_d are the signal and noise functions correspondingly, calculated by

$$S_d(L_{\min}, L_{\max}, D_{\text{win}}) = P_s^0 \times \sum_{i=L_{\min}}^M (E_s^r) \quad (18)$$

$$N_d(L_{\min}, L_{\max}, D_{\text{win}}) = \sum_{i=L_{\min}}^M (d_0 - E_s^r). \quad (19)$$

As a result of this analysis, we conclude that the optimal distance is a function of L_{\min} , L_{\max} and should be considered as an independent parameter. In order to evaluate our model, we will repeat the extraction of correlation between the SN_d and actual performance for different values of D_{win} .

Indeed, SN_d offers an important 0.920 rank correlation (Spearman) to the maximum performance that can be achieved for the selected optimal n-gram range (see Figure 10). SN_d has a promising Pearson correlation of 0.896 to the performance. Therefore, our estimator is good in finding near-optimal D_{win} values.¹¹ In the given example of the DUC 2006 corpus, the best performance

¹¹This holds for already decided optimal values of L_{\min} , L_{\max} we found through a series of experiments.

Table VI. Correlation of Measures to the Content Responsiveness Metric of DUC 2006 for Automatic Peers Only. Within Parentheses, the p-value of the Corresponding Test

| Metric | Spearman | Pearson | Kendall |
|------------------------|--------------|--------------|--------------|
| Overall Responsiveness | 0.718 | 0.833 | |
| Rouge-2 | 0.767 | 0.836 | |
| Rouge-SU4 | 0.790 | 0.850 | |
| BE-HM | 0.797 | 0.782 | |
| <i>AutoSummENG</i> | 0.870 (0.00) | 0.904 (0.00) | 0.712 (0.00) |
| <i>AutoSummENG B/S</i> | 0.858 (0.00) | 0.899 (0.00) | 0.712 (0.00) |

was 0.938 and the one returned using the estimation was 0.935, while the average over all candidate distances was 0.927 with a standard deviation of 0.008.

The near-optimal values for the pair (L_{\min}, L_{\max}) were according to the estimation values (4, 4), while indeed the optimal values were (1, 3) (DUC 2006 corpus).

5. OVERALL PERFORMANCE OF AUTOSUMMENG

In order to check the performance of our method compared to other existing methods, we used the Spearman correlation and Pearson correlation that is used in Dang [2006], but for our method we have also calculated the Kendall's tau correlation coefficient, which we consider to be the most fitting coefficient for the given task based on its definition (see Section 2.1). The Responsiveness measure in DUC 2006 was named Content Responsiveness because another measure appeared named Overall Responsiveness (see Dang [2006]). Briefly, Overall Responsiveness represents an overall quality measure (including grammaticality and other textual qualities) for a given system, while Content Responsiveness only refers to whether the required information was contained in summaries from a given system without taking into account the well-formedness of output summary. The results concern application of the character n-gram Graph—Value representation with a symmetric window. P-values reported zero (0.00) indicate actual p-values that are rounded to zero when two digits are considered significant. We should note that we have also used an ordinary nonparametric bootstrapping approach¹² with 10,000 replications to better determine the results for our method. The corresponding results appear either in the AutoSummENG entries as separate entries (see Table VI) or as second value—p-value pairs (see Table VII).¹³

In Table VI, there is an obvious difference between the performance of AutoSummENG and existing approaches. Table VII indicates the Pearson correlation performance of evaluation methods when not including system 17 of DUC 2006 for which BE-HM breaks (due to some characters in the input) and performs abnormally.

¹²For an introduction to bootstrapping see Efron and Tibshirani [1993].

¹³Given the fact that the results using bootstrapping were only marginally modified over many experiments, we did not further perform bootstrapping. Since the original values were considered good and indicative estimations of the process, we did not want to diversify the method of calculation of our results from other corresponding works.

Table VII. Pearson Correlation of Measures to the Content Responsiveness Metric of DUC 2006 for Automatic Peers, Human Peers and All Peers, Excluding Peer 17. Within Parentheses, the p-value of the Corresponding Test

| Evaluated Group | Rouge-2 | Rouge-SU4 | BE-HM | AutoSummENG – B/S |
|-----------------|-------------|-------------|-------------|-------------------------|
| Automatic Peers | 0.84 (0.00) | 0.85 (0.00) | 0.78 (0.00) | 0.91 (0.00)–0.90 (0.00) |
| Human Peers | 0.64 (0.05) | 0.69 (0.03) | 0.57 (0.09) | 0.68 (0.03)–0.67 (0.00) |
| All Peers | 0.90 (0.00) | 0.88 (0.00) | 0.88 (0.00) | 0.97 (0.00)–0.97 (0.00) |

Table VIII. Correlation of AutoSummENG to the Responsiveness Metric of DUC 2005 for Automatic Peers, Human Peers and All Peers. Within Parentheses, the p-value of the Corresponding Test

| Evaluated Group | Spearman | Pearson | Kendall |
|-----------------|--------------|--------------|--------------|
| Automatic peers | 0.906 (0.00) | 0.908 (0.00) | 0.755 (0.00) |
| Human peers | 0.857 (0.00) | 0.830 (0.00) | 0.764 (0.00) |
| All peers | 0.957 (0.00) | 0.985 (0.00) | 0.847 (0.00) |

Table IX. Correlation of AutoSummENG to the Content Responsiveness Metric of DUC 2006 for Automatic Peers, Human Peers and All Peers. Within Parentheses, the p-value of the Corresponding Test. Statistical Importance Lower Than the 95% Threshold are Noted by Emphatic Text in the Parentheses

| Evaluated Group | Spearman | Pearson | Kendall |
|-----------------|--------------|--------------|--------------|
| Automatic peers | 0.870 (0.00) | 0.904 (0.00) | 0.712 (0.00) |
| Human peers | 0.648 (0.04) | 0.684 (0.03) | 0.471 (0.07) |
| All peers | 0.935 (0.00) | 0.966 (0.00) | 0.804 (0.00) |

As an overview of the major evaluation systems' performance over the data of DUC 2005 to 2007, Table XI has been provided, based partly on Conroy and Dang [2008]. It should be noted that the AutoSummENG performance does not correspond necessarily to its optimal value but rather to the performance achieved using preestimated parameters. Another important note is that, even though there is a difference between the performance of systems, statistical analysis indicates through confidence intervals that the difference in performance may be due to randomness (see also Dang [2005, 2006] and Hovy et al. [2005]).

Even though the evaluation process itself contains the a-priori estimation step for its parameters, we wanted to check whether the model parameters determined for the corpus of DUC 2006 would function effectively when applied to DUC 2005 and DUC 2007 corpora. In Tables VIII, IX, X, we can see the results for all corpora (DUC 2005, DUC 2006, DUC 2007). In the tables, the results have been separated by groups (automatic peers and human peers), and there is also the overall ranking correlation, including all peers. The results indicate that the DUC 2006 parameters perform well in other corpora as well, showing that the parameters did not simply overfit the DUC 2006 corpus.

To verify this fact, we also determined model parameters for DUC 2005 and applied them to all corpora: DUC 2005, DUC 2006, DUC 2007. The results were once more satisfying as can be seen in Table XII. This hints at the fact that the

Table X. Correlation of AutoSummENG to the Content Responsiveness Metric of DUC 2007 for Automatic Peers, Human Peers and All Peers. Within Parentheses, the p-value of the Corresponding Test. Statistical Importance Lower Than the 95% Threshold is Noted by Emphatic Text in the Parentheses

| Evaluated Group | Spearman | Pearson | Kendall |
|-----------------|--------------|--------------|--------------|
| Automatic peers | 0.864 (0.00) | 0.88 (0.00) | 0.707 (0.00) |
| Human peers | 0.615 (0.06) | 0.649 (0.04) | 0.396 (0.12) |
| All peers | 0.935 (0.00) | 0.964 (0.00) | 0.801 (0.00) |

Table XI. Pearson Correlation of Measures to the Content Responsiveness Metric of DUC 2005–2007 for Automatic Systems

| Year | BE(-HM) | Rouge-2 | Rouge-SU4 | AutoSummENG |
|------|---------|---------|-----------|-------------|
| 2005 | 0.87 | 0.94 | 0.93 | 0.91 |
| 2006 | 0.85 | 0.84 | 0.85 | 0.90 |
| 2007 | 0.89 | 0.88 | 0.83 | 0.88 |

model parameters are more language-dependent than corpus-dependent, but this will have to be verified against another language.

The fact that our method does not require parsing of some kind, nor syntactic or grammatical analysis like other existing methods, offers an advantage, both in terms of complexity as well as in terms of inherited noise from erroneous preprocessing (which was indicated as a problem in the case of BE [Dang 2006]).

In the course of our experiments, we used the optimal values found for content responsiveness correlation to check the correlation of the proposed method to the overall responsiveness of systems in DUC 2006. The results are illustrated in Table XIII. Once more the method seems to do perform adequately with the histogram-cooccurrence version reaching the highest performance (for the given parameter setting of $L_{\min} = 4$, $L_{\max} = 4$, $D_{\text{win}} = 4$). This indicates that our method can have more applications than meets the eye and this seems worth investigating.

On the negative side of this analysis, the fact that we need to estimate parameters can be time-consuming and even error-prone, which will affect the overall performance of the system. This problem is only partially addressed by the robustness of the results for nonmarginal parameter values. Another lesser drawback of our method is that the graph representation can be memory-consuming even though in our implementation¹⁴ we have optimized the code and the problem has been tackled. Finally, there have been no experiments in different languages, which means that we have not answered the question of whether the language-neutral approach will have similar performance in other languages. On the other hand, this does not contradict the fact that the approach remains strictly language-neutral in its methodology.

¹⁴The full project, including source code, of AutoSummENG, can be found at <http://www.ontosum.org/static/AutomaticSummarization>.

Table XII. Correlation of AutoSummENG to the Responsiveness Metric of DUC 2005 and Content Responsiveness Metric of DUC 2006, 2007 for Automatic Peers, Human Peers and All Peers Using Estimated Parameters Based on DUC 2005. Within Parentheses, the p-value of the Corresponding Test. Statistical Importance Lower Than the 95% Threshold are Noted by Emphatic Text in the Parentheses

| Year - Evaluated Group | Spearman | Pearson | Kendall |
|------------------------|--------------|--------------|--------------|
| 2005 - Automatic peers | 0.840 (0.0) | 0.885 (0.0) | 0.669 (0.0) |
| 2005 - Human peers | 0.936 (0.0) | 0.878 (0.00) | 0.854 (0.00) |
| 2005 - All peers | 0.929 (0.00) | 0.977 (0.00) | 0.803 (0.0) |
| 2006 - Automatic peers | 0.871 (0.0) | 0.891 (0.0) | 0.709 (0.0) |
| 2006 - Human peers | 0.759 (0.01) | 0.715 (0.02) | 0.566 (0.03) |
| 2006 - All peers | 0.937 (0.00) | 0.967 (0.00) | 0.806 (0.0) |
| 2007 - Automatic peers | 0.842 (0.0) | 0.871 (0.0) | 0.687 (0.0) |
| 2007 - Human peers | 0.659 (0.04) | 0.673 (0.03) | 0.442 (0.08) |
| 2007 - All peers | 0.925 (0.00) | 0.966 (0.00) | 0.792 (0.0) |

Table XIII. Correlation of AutoSummENG to the Overall Responsiveness Metric G Stands for Graph, H for Histogram, C for Cooccurrence and V for Value

| Representation | Spearman | Pearson |
|----------------|----------|---------|
| G - C | 0.748 | 0.860 |
| G - V | 0.786 | 0.893 |
| H - C | 0.811 | 0.920 |
| H - V | 0.537 | 0.858 |

6. CONCLUSIONS—FUTURE WORK

The AutoSummENG method for summarization evaluation is a promising method based on language-neutral analysis of texts and comparison to gold-standard summaries. The method is based on n-gram graphs even though it provides support for other histogram-based approaches. We have found that the character n-gram graph representation, including information of neighborhood frequency, can render the best results for the given task. The presented method appears to outperform current approaches in the corpus of DUC 2006, providing a good evaluation alternative for future attempts.

Answering the questions posed in Section 1.1, statistical information related to cooccurrence of character n-grams seems to provide important information concerning the evaluation process of summary systems. As described within our method, the actual representation used for capturing this information can be an n-gram graph with parameters optimized a priori. The distance metric preferred is the value similarity between the graph representation of peer summaries and model summaries. Our method, complemented by the parameter optimization step, has proved to be a language-neutral, fully automated, context-sensitive method with competitive performance.

Concerning future work, it would be interesting to use only symbols in our analysis of the summary texts to see if there is hidden information in what we have called non-symbols. On the same basis, a parameter-free version of the method would determine n-grams of various length and synthesize a single

n-gram graph not requiring any input but the corpus of summaries. It would also be interesting to investigate different types of neighborhood and different functions of importance for neighborhood as well as different weighting functions for the importance of matching n-grams of specific rank (e.g., longer vs. shorter n-grams) or nature (e.g., rare vs. common n-grams). We are planning to investigate the evaluation of other characteristics, like grammaticality, cohesion, and so forth, using the same scheme with different parameters. In this investigation, the word n-gram dimension should be reexamined because it may provide more noise-free information considering the fact that whole words usually follow our definition of symbols by being meaningful. Furthermore, there is a drawback to our method tied to the complexity of extracting and comparing n-gram graphs. This drawback has already been handled in terms of implementation, but the algorithm itself holds a complexity much higher than that of constructing a histogram per se. Therefore, it would be interesting to hold only useful subgraphs based on a statistical measure of usefulness or find an algorithmic alternative to our own.

Even though a series of numerous experiments has already been conducted, we feel that the presented method should be further investigated as it may hold implications concerning the potential for using the graph representation in a language-neutral way for various NLP tasks.

ACKNOWLEDGMENTS

The corpora of DUC 2005, 2006, 2007 were kindly provided by NIST and has been used according to the TIPSTER, TREC directions. The whole range of experiments for this article as well as the editing process and statistical analysis were conducted on Open Source software. The JInsect toolbox is implemented with Java programming language and can be found as an open source project at <http://www.ontosum.org/static/AutomaticSummarization>.

REFERENCES

- ANDERSON, J. 1985. *Cognitive Psychology and Its Implications*, 2nd ed. WH Freeman.
- BANKO, M. AND VANDERWENDE, L. 2004. Using n-grams to understand the nature of summaries. In *HLT-NAACL 2004: Short Papers*, D. M. Susan Dumais and S. Roukos, Eds. Association for Computational Linguistics, Boston, Massachusetts, USA, 1–4.
- BUNKE, H. 1998. Error-tolerant graph matching: A formal framework and algorithms. *Advances in Pattern Recognition, LNCS 1451*, 1–14.
- CAVNAR, W. B. AND TRENKLE, J. M. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*. Las Vegas, US, 161–175.
- CLEVELAND, W. 1981. LOWESS: A program for smoothing scatterplots by robust locally weighted regression. *The American Statistician* 35, 1, 54.
- CONROY, J. AND DANG, H. T. 2008. Mind the gap: dangers of divorcing evaluations of summary content from linguistic quality. In *Coling 2008* (to appear).
- COPECK, T. AND SZPAKOWICZ, S. 2004. Vocabulary Usage in Newswire Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. Association for Computational Linguistics, 19–26.
- DA SILVA, J., DIAS, G., GUILLORÉ, S., AND LOPES, J. 1999. Using LocalMaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. *Proceedings of the 9th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence*, 113–132.

- DANG, H. 2005. Overview of DUC 2005. In *Proceedings of the Document Understanding Conference Workshop (DUC'05) at the Human Language Technology Conference / Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP'05)*.
- DANG, H. 2006. Overview of DUC 2006. In *Proceedings of HLT-NAACL'06*.
- DUDA, R., HART, P., AND STORK, D. 2001. *Pattern Classification*. John Wiley New York, NY.
- EFRON, B. AND TIBSHIRANI, R. 1993. *An Introduction to the Bootstrap*. Chapman & Hall/CRC.
- ENDRES-NIGGEMEYER, B. 2000. Human-style www summarization. <http://endres.niggemeyer.fh-hannover.de/paps.html>
- ERKAN, G. AND RADEV, D. 2004a. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Art. Int. Resear.* 22, 457–479.
- ERKAN, G. AND RADEV, D. 2004b. Michigan at DUC 2004—Using sentence prestige for document summarization. In *Proceedings of the Document Understanding Conferences*.
- GIANNAKOPOULOS, G., KARKALETIS, V., AND VOUIROS, G. 2006. Automatic Multi-document summarization and prior knowledge: Past, present and vision. Tech. rep. (DEMO-2006-2). NCSR Demokritos.
- HOLLANDER, M. AND WOLFE, D. 1973. *Nonparametric Statistical Inference*. John Wiley, New York, NY.
- HOUVARDAS, J. AND STAMATATOS, E. 2006. N-gram feature selection for authorship identification. In *Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA'06)*. 77–86.
- HOVY, E., LIN, C., AND ZHOU, L. 2005a. Evaluating DUC 2005 using basic elements. In *Proceedings of the Document Understanding Conference (DUC'05)*.
- HOVY, E., LIN, C., ZHOU, L., AND FUKUMOTO, J. 2005b. Basic Elements. <http://haydn.isi.edu/BE/>.
- HOVY, E., LIN, C., ZHOU, L., AND FUKUMOTO, J. 2006. Automated summarization evaluation with basic elements. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC)*.
- KENDALL, M. 1962. *Rank Correlation Methods*. Hafner, New York, NY.
- LAMKHEDE, S. 2005. Multidocument summarization using concept chain graphs. M.S. thesis, State University of New York at Buffalo.
- LIN, C. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS'04)*. 25–26.
- LIN, C. AND HOVY, E. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL Workshop on Automatic Summarization*. Vol. 4. Association for Computational Linguistics Morristown, NJ, 45–51.
- LIN, C.-Y. AND HOVY, E. 2003. Automatic evaluation of summaries using n-gram cooccurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL'03)*. Association for Computational Linguistics, Morristown, NJ, 71–78.
- LUHN, H. 1958. The automatic creation of literature abstracts. *IBM J. Resear. Devel.* 2, 2, 159–165.
- MANI, I. AND BLOEDORN, E. 1997. Multi-document summarization by graph search and matching. In *Proceedings of AAAI*. AAAI, 622–628.
- MANNING, C. AND SCHÜTZE, H. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.
- MARCU, D. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press.
- MASSEY JR, F. 1951. The Kolmogorov-Smirnov test for goodness of fit. *J. Amer. Statist. Ass.* 46, 253, 68–78.
- MATSUO, Y., OHSAWA, Y., AND ISHIZUKA, M. 2001. A Document as a Small World. In *Proceedings the 5th World Multi-Conference on Systemics, Cybenetics and Infomatics (SCF'01)* Vol. 8. 410–414.
- MIHALCEA, R. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*. ACL.
- MIHALCEA, R. 2005. Multi-document Summarization with iterative graph-based algorithms. In *Proceedings of the 1st International Conference on Intelligent Analysis Methods and Tools (IA'05)*.

- MOHAMED, A. AND RAJASEKARAN, S. 2006. Query-based summarization based on document graphs. In *Proceedings of the Document Understanding Conference (DUC'06)*.
- NAVARRO, G. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.* 33, 1, 31–88.
- NENKOVA, A. 2006. Understanding the process of multi-document summarization: Content selection, rewrite, evaluation. Ph.D. thesis, Columbia University.
- OTTERBACHER, J., ERKAN, G., AND RADEV, D. R. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT'05)*. Association for Computational Linguistics, Morristown, NJ, 915–922.
- PAPINENI, K., ROUKOS, S., WARD, T., AND ZHU, W. 2001. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 311–318.
- PASSONNEAU, R., MCKEOWN, K., SIGELMAN, S., AND GOODKIND, A. 2006. Applying the Pyramid method. In *Proceedings of the Document Understanding Conference (DUC'06)*.
- RADEV, D. 2000. A common theory of information fusion from multiple text sources, step one: Cross-document structure. In *Proceedings of the 1st ACL SIGDIAL Workshop on Discourse and Dialogue*.
- RADEV, D., JING, H., AND BUDZIKOWSKA, M. 2000. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the ANLP/NAACL Workshop on Summarization*.
- RAYMOND, J., GARDINER, E., AND WILLETT, P. 2002. RASCAL: Calculation of graph similarity using maximum common edge subgraphs. *Comput. J.* 45, 6, 631.
- SAGGION, H. AND LAPALME, G. 2002. Generating indicative-informative summaries with sumUM. *Comput. Lingu.* 28, 4, 497–526.
- SPEARMAN, C. 1906. Footrule for measuring correlation. *British J. Psych.* 2, 89–108.
- STEPHENS, M. 1974. EDF Statistics for goodness of fit and some comparisons. *J. Amer. Statist. Ass.* 69, 347, 730–737.
- UKKONEN, E. 1992. Approximate string-matching with q-grams and maximal matches. *Theor. Comput. Sci.* 92, 1, 191–211.
- VAN HALTEREN, H. AND TEUFEL, S. 2003. Examining the consensus between human summaries: Initial experiments with factoid analysis. In *Proceedings of the HLT-NAACL Workshop on Text Summarization*. Vol. 5. Association for Computational Linguistics Morristown, NJ, 57–64.
- VOORHEES, E. 2003. Overview of the TREC 2003 question answering track. In *Proceedings of the 12th Text REtrieval Conference (TREC'03)*.
- WITTE, R., KRESTEL, R., AND BERGLER, S. 2006. Context-based multi-document summarization using fuzzy coreference cluster graphs. In *Proceedings of Document Understanding Workshop (DUC)*. New York, NY.
- ZENS, R. AND NEY, H. 2006. N-gram posterior probabilities for statistical machine translation. In *Proceedings of the Human Language Technology Conference Workshop on Statistical Machine Translation*. 72–77.
- ZHOU, L., LIN, C., MUNTEANU, D., AND HOVY, E. 2006. ParaEval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the Human Language Technology Conference North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL'06)*.

Received January 2008; revised May 2008; accepted June 2008