Preprocessing and Term Weights in Multilingual Summarization

John M. Conroy
IDA / Center for Computing Sciences
conroy@super.org

Sashka T. Davis
IDA / Center for Computing Sciences
stdavi3@super.org

Jeff Kubina
Department of Defense
jmkubin@tycho.ncsc.mil

Abstract

In this paper we examine the performance of four term weighting approaches for multilingual single and multi-document summarization and the effects various tokenization and sentence splitting methods have on their performance. We introduce a new single document summarization method that only uses the document's hierarchical structure to compute a summary. Finally we present the performance results of each method on the SIGDIAL 2015 Multilingual Single and Multi-Document Summarization Tasks.

1 Single and Multi-Document Summarization

Developing techniques to perform multilingual summarization across many languages presents many challenges. In this paper we consider alternatives to two major challenges in multilingual summarization. The first is accurate tokenization and sentence boundary detection of a document. The SIGDIAL 2015 Multilingual Single and Multi-Document Summarization Tasks, denoted by MSS and MMS respectively, enabled us to compare our multilingual sentence splitter FASST (Fast, Accurate, Sentence Splitter for Text) against the performance of Basis Technology's natural language processing package called Rosette (Basis Technology, 2015) and the Natural Language Toolkit (Bird et al., 2009).

The second major challenge we study is term weighting. Term weights are used as input into our method for language independent extractive summarization algorithm called OCCAMS (Optimal Combinatorial Covering Algorithm for Multi-document Summarization). In Davis et al. (2012) we show that given oracle weights OCCAMS produces summaries which significantly outscore human summarizers in both coverage scores and ROUGE scores. In this paper we aim to close the gap between the performance of the oracle weighting and our term-weighting and to better approximate an oracle weighting (Conroy et al., 2006). We will explore the performance of: term-frequency weighting (TF), personalized term rank, and nonnegative matrix factorization (Conroy et al., 2013) techniques for term weighting.

By evaluating the capabilities of the different linquistic techniques for sentence splitting and tokenization, and the different approaches for term-weighing, we hope to close the performance gap between our summarization systems and the OCCAMS summarizer given oracle weights.

Our method for computing a summary of a document consists of the following steps: *1*) Sentence splitting *2*) tokenization and language specific lemmatization or stemming *3*) forming a term-sentence matrix *4*) term weight generation *5*) sentence selection *6*) sentence ordering. The first two steps, sentence splitting and tokenization, are language dependent, but the remaining steps are language independent. Next we briefly discuss each of these steps but will focus on sentence splitting and tokenization as they are a focus of our major experiments for MultiLing 2015.

2 Sentence Boundary Detection and Tokenization

We desired to test the hypothesis as to whether or not more sophisticated sentence splitters and tokenizers can improve the ability of a text summarizer to generate a higher quality summary. In particular, how well does our rule based sentence splitter FASST-E (Conroy et al., 2009) and its multilingual extensions (Conroy et al., 2011) used in conjunction with regular expression for tokenization compare with more sophisticated systems. We consider Basis Technology's natural language processing package called Rosette and the Natural Language Toolkit, denoted NLTK, that are known to use the Unicode standard breaking algorithm (The Unicode Consortium, 2014), linguistic analysis, statistical modeling, and machine learning to perform sentence splitting and lemmatization as part of the tokenization. Proper *tokenization* can improve the quality of the summary. FASST-E simply employs the Porter stemmer whereas Rosette performs lemmatization and uses morphological analysis to disambiguate compound words in languages such as Arabic or German.

2.1 Rosette and NTK

Rosette was used to sentence split, tokenize, and, when available, lemmatize Arabic, Chinese, Czech, Dutch, English, French, German, Greek, Hungarian, Italian, Japanese, Korean, Norwegian (used Norwegian-nynorsk), Norwegian-bokmal, Norwegian-nynorsk, Polish, Romanian, Russian, Spanish, Swedish, Thai, and Turkish.

Rosette did not have a language processor for Africaans and Bulgarian, so for those two cases we used the closest matching language processor, Dutch for Africaans and Russian for Bulgarian.

The NLTK was used to sentence split and tokenize Basque, Catalan, Croatian, Esperanto, Finnish, Georgian, Indonesia, Malay, Persian, Portuguese, Serbian, Serbo-croatian, Slovak, Slovene, and Vietnamese.

2.2 FASST

FASST uses a series of linguistic cues to split sentences which are then formulated as regular expressions. In addition for English we perfrormed additional processing to trim the sentence produced by FASST. The sentence trimmer is implemented as a series of regular expressions and a detailed description of it can be found in Conroy et al. (2009). The regular expressions used for tokenization of sentence (declared by FASST), largely break the sentence into either white space delimited tokens, devoid of punctuation that are then stemmed in the case of English text.

3 Term-Sentence Matrices

Term weights for indicating the relative importance of a term are computed based on a term-sentence matrix, which is an instance of the vector space model. This model was introduced by Salton (1991) and shortly after Dumais (1994) proposed dimensionality reduction in document retrieval systems, which has been used by many other researchers for document summarization. We construct the term-sentence matrix $A = (a_{i,j})$, where $i = 1, \ldots, m$ for the terms, and $j = 1, \ldots, n$ for the valid sentences discovered in the document for MSS or a collection of documents for MMS. The column labels are the sentences $S_1, \ldots, S_n$ of the document(s), while the row labels of the term-sentence matrix are the terms $T = (t_1, \ldots, t_m)$ The matrix A is defined by

$$a_{i,j} = \ell_{i,j}$$

where, $\ell_{i,j}$ is 1 when term i appears in sentence j and 0 otherwise.

The set of terms, $T = (t_1, \ldots, t_m)$ are selected by using Dunning's likelihood statistic (Dunning, 1993), which is equivalent to a mutual information statistic. The background used in the MSS task was the collection of Wikipedia feature articles in the given language. For the MMS task, the background used was simply the other 9 documents in the MMS data set for the language. The rejection threshold (p-value) was chosen to insure that the number of terms in the matrix A was at least twice the target length of the summary.

4 Term Weights

Given the term-sentence matrix as defined in the previous section we estimate the relative importance of a term, which is designed to approximate the probability that a human would include a term in a summary (Conroy et al., 2006). For MultiLing15 we considered four approaches for approximating the oracle score: term-frequency (TF), personalized term-rank, and two based on nonnegative matrix factorization methods.

4.1 Term Frequency

Term frequency and its variants is a commonly used term-weight for summarization, information retrieval, and keyword identification tasks. The term frequency f_i of term i, is defined as the number of times i appears in all sentences.

4.2 Personalize Term Rank

The personalized term rank (PTR) we used in Conroy et al. (2013), is a variant of TextRank Mihalcea (2005), that uses the high mutual information terms in the computation of the term rank. The personalization vector is simply the normalized term frequency. Thus the resulting stationary vector will reflect not only the frequency of the terms in the document but also the co-occurrence of the terms in the sentences.

4.3 Nonnegative Matrix Factorization

Nonnegative matrix factorization (NNMF) of a term sentence matrix can be used for dimensionality reduction and as such is another alternative to improve over term frequency (Conroy et al., 2013). NNMF, much like the method of latent semantics analysis, requires a selection of the dimension k for the rank approximation of the term-by-sentence matrix. We used the method of alternating least squares to compute an approximate factorization $A \approx WH$, where W has k columns and H has k row. In this paper we used Matlab's function *nnmf()*, with 20 random starts to improve the approximation. The dimension k was chosen for the MSS and MMS based on experiments with the training data. The term-weights given to OCCAMS are simply the row sums of WH.

4.4 Interval Bounded Nonnegative Matrix Factorization (IBNMF)

The fourth term-weighting method we explore uses the interval bounded nonnegative matrix factorization (IBNMF), first introduced in Conroy et al. (2013). Here in this paper we use the output of NNMF from 20 random starts as initial input to IBMNF. We use the sums of the rows of the resulting factorization, as was the case for NNMF, to estimate the term weights.

5 Sentence Selection with OCCAMS

The OCCAMS algorithm for extractive summarization chooses a set of sentences whose combined weight of terms is maximized while the combined sum of lengths of the sentence selected must not exceed the bound on the size of the summary L. OCCAMS takes as input a set of sentences, where each sentence is a set of terms. Each term has a weight and each sentence has a length. OCCAMS outputs a set of sentences that maximize the combined term weights and minimizes redunancy, which is especially important for the multi-document summarization task. Here in this paper we use Conroy et al. (2013) version of OCCAMS. Davis et al. (2012) contains detailed description and combinatorial analysis of the performance of the algorithm.

Sentence ordering is done by employing an approximate traveling salesperson algorithm (Conroy et al., 2009).

6 Hierarchical Sentence Interweaving

To assess how much the structure of a large document alone can contribute to generating a summary, we developed an algorithm that computes a summary of a document using only the tree structure of the document's sections, paragraphs, and sentences. The algorithm computes a summary by hierarchically interweaving the sentences of the paragraphs and sections of the document and does not perform any statistical analysis of the document, or the dataset. We were able to evaluate the algorithm only on the MSS dataset because the XML version of the test documents preserved the hierarchical structure of the sections and paragraphs. Depending on the language the sentences of the paragraphs were identified using Rosette or the NLTK. Figure 1 illustrates how the sentences of a document are hierarchically interweaved. The sentences of paragraphs in a section are first interweaved by their positions. At higher section levels the subsequent list of sentences are interweaved and the summary is obtained from the top list of sentences truncated to the desired size. The algorithm selects the sentences that a human reader would read when skimming a large, well written, document.

6.1 Multi-Document Summarization

For the MMS task the main goal was to compare the FASST sentence splitting (plus the Porter stemmer and FASST trimmer for English) to the combination of Rosette and the Python NLTK, where lemmatization was available in a number of languages in addition to the sentence splitting. Table 1 gives the system mapping of the three entries.

The ROUGE-2 and ROUGE-4 evaluations for MMS are illustrated in Figures 2 and 3. The scores show the systems are competitive and indeed that Basis and NLTK-TF out-perform FASST-TF in all but Chinese and Arabic. Although individually these results are not significant, in conjunction

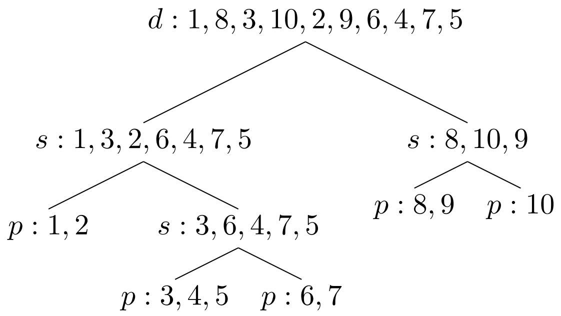

Figure 1: In the document tree the leaves are the paragraphs, labeled with p, with their list of sentences represented as numbers. At each section level, labeled with s, the list of sentences in the subsections or paragraphs are interweaved. The summary for the document is obtained by truncating the final weaving of the sentences in the document, labeled with d, to the appropriate size.

Label	Method
MMS8a	FASST TF
MMS8b	Basis/NLTK TF
MMS8c	FASST IBNMF 24

Table 1: Labels of systems submitted for the MMS task.

they give strong evidence that Basis and NLTK are improving the summaries generated.

6.2 Results for the MSS at MultiLing 2015

In the multilingual single document task (MSS) the four term weighting approaches were used as well as the hierarchical sentence interweaving. Table 2 contains the five summary methods we submitted to the MSS task.

A total of 22 systems participated in the MSS task. The CCS entries and two other systems were only ones to submit summaries for all 38 languages. The maximum number of languages submitted by other systems was three. A lead summary was used as the task baseline. All systems were scored using ROUGE-1, 2, 3, and 4 (Lin, 2004) as well as MeMoG (Giannakopoulos et al., 2008; Giannakopoulos et al., 2010). A non-parametric analysis of variance test was used to measure any significance difference between the systems. The last row of Table 3 gives the fraction of languages (out of 38) where a significant difference was observed by the ANOVA for ROUGE-2, 4 and MeMoG. Each other row gives the fraction of times that a system significantly outperformed (as measured by a paired Wilcoxon test) the lead baseline. It is worthy to note that the hierarchical sentence interweaving method, CCS5, is the system that most often significantly outperforms the baseline.

Label	Method
CCS1	IBNMF Rank 25
CCS2	NMF Rank 25
CCS3	PTR
CCS4	TF
CCS5	Interweaving

Table 2: Labels of systems submitted for the MSS task.

System	ROUGE-2	ROUGE-4	MeMoG
CCS1	20/38	3/38	19/38
CCS2	21/38	4/38	19/38
CCS3	21/38	3/38	19/38
CCS4	20/38	2/38	20/38
CCS5	23/38	7/38	20/38
EXB1	15/38	1/38	11/38
LCS-IESI1	6/38	2/38	6/38
ANOVA	25/38	10/38	21/38

Table 3: The first seven rows in the table provide the fraction of times that the corresponding system significantly outperformed the lead baseline. The last row gives the fraction of languages (out of 38) where a significant difference was observed by the ANOVA.

Figure 4 gives a scatter plot of ROUGE-3 scores for the language where the ANOVA indicated a significant difference among the system.

Finally, we turn to comparing the five CCS systems. Table 4 is a matrix giving the number of languages that CCS system i significantly out scored system j in ROUGE-4. The table clearly shows that there are few significant differences within the first four systems, indicating that the term weighting are all about as good as TF on the high mutual information bigrams. Whereas the hierarchical sentence interweaving significantly out performs the term-weighting methods in about $1/4$ of the languages.

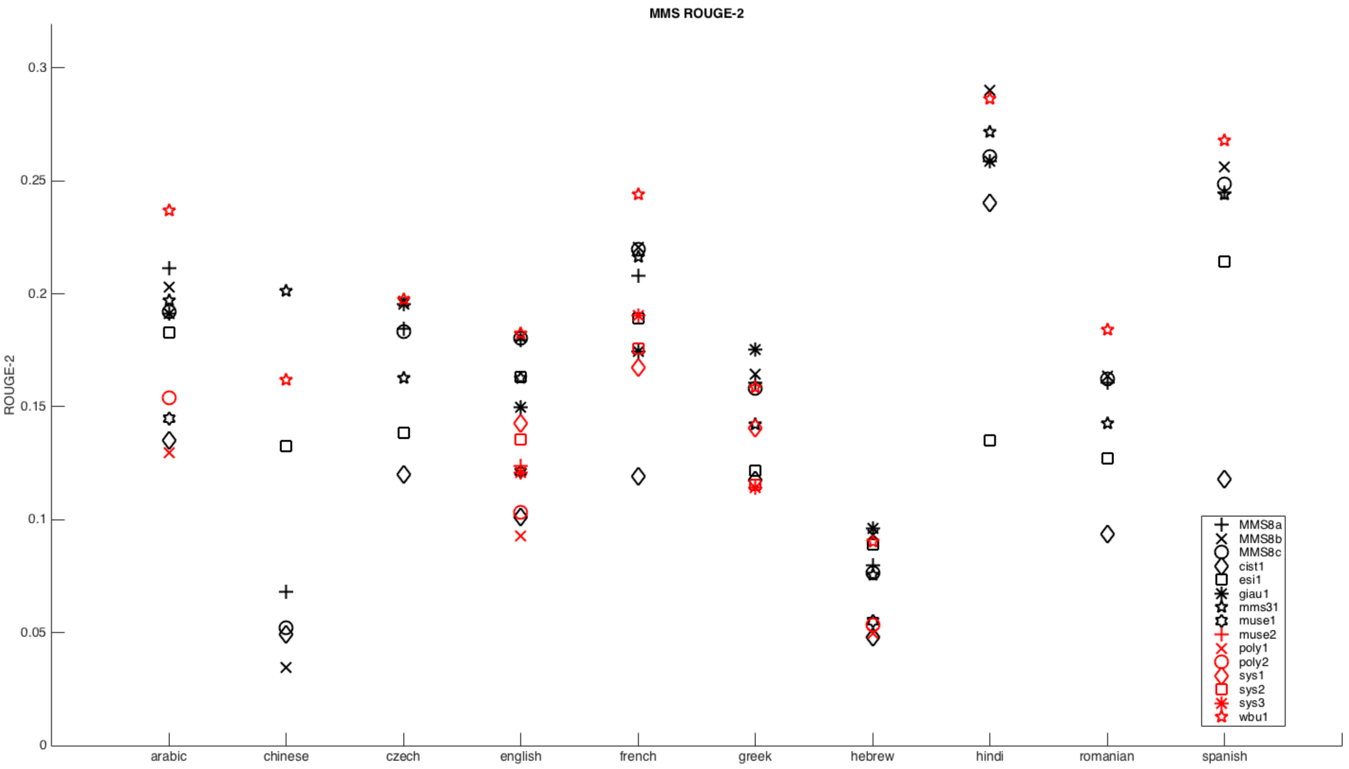

Figure 2: ROUGE-2 scores for MMS

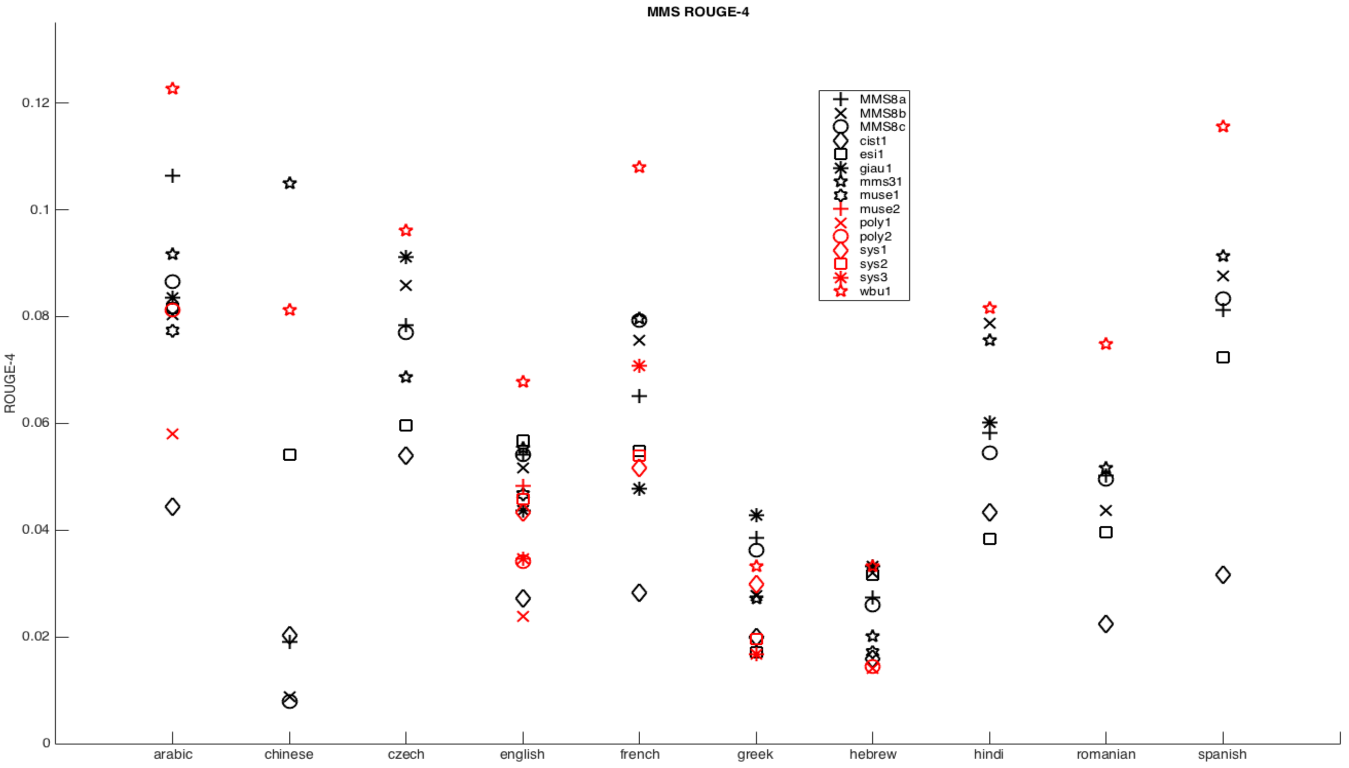

Figure 3: ROUGE-4 scores for MMS

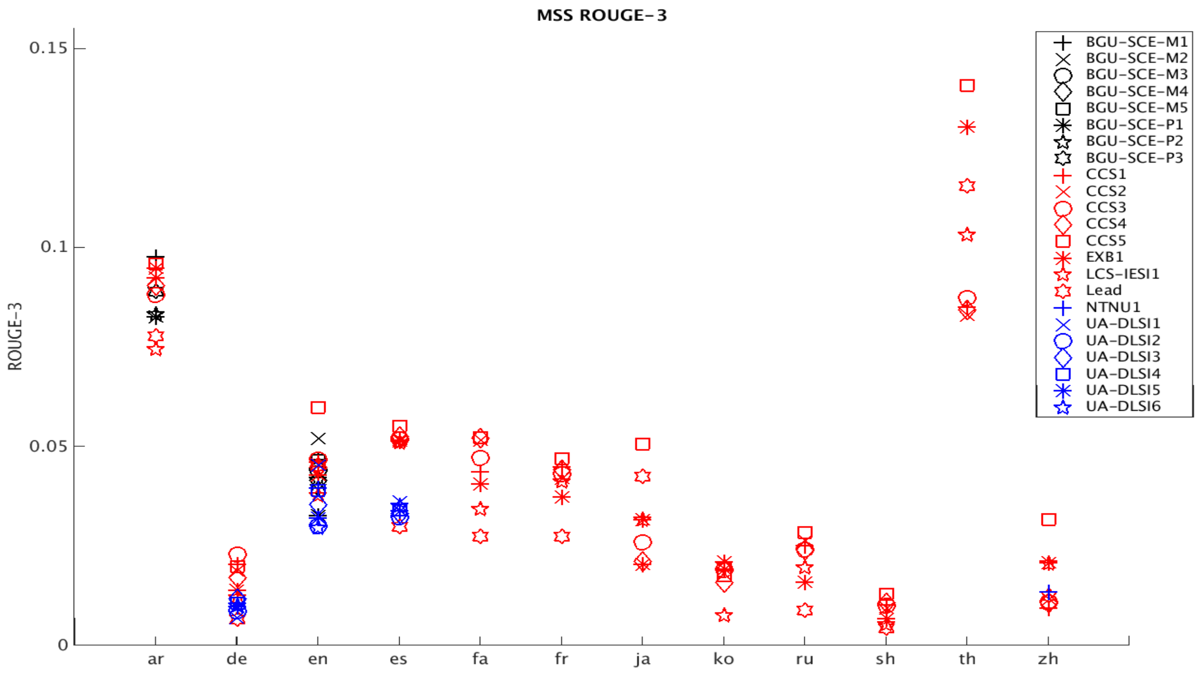

Figure 4: ROUGE-3 scores for MSS

	CCS1	CCS2	CCS3	CCS4	CCS5
CCS1	0	0	1	0	0
CCS2	1	0	2	2	0
CCS3	1	1	0	2	0
CCS4	2	2	0	0	0
CCS5	8	9	10	8	0

Table 4: The table shows the number of times that each system significantly outperformed another as returned by a pair Wilcoxon test using the ROUGE-4 scores.

7 Conclusions and Future Work

In this paper we compared the rule base FASST sentence splitter and the Rosette and NLTK sentence splitters and tokenizers. The differences on the MMS task were small but largely consistently in favor of the Rosette and NLTK. No significant gain was seen in the use of NMF for the MMS task.

On the MSS task the four term-weighting methods with OCCAMS sentence selection showed great promise, significantly outperforming the lead baseline more than other systems in the task. The best performer appears to be the hierarchical sentence interweaving method. This system did not use OCCAMS and in future work the two systems will be combine.

References

Basis Technology. 2015. Rosette base linguistics (rbl-je) version 7.12.0. http://www.basistech.com.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the ACL'06/COLING'06 Conferences*, pages 152–159, Sydney, Australia, July.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2009. CLASSY 2009: summarization and metrics. *Proceedings of the text analysis conference (TAC)*.

John M. Conroy, Judith D. Schlesinger, Jeff Kubina, Peter A. Rankel, and Dianne P. O'Leary. 2011. CLASSY 2011 at TAC: Guided and multi-lingual summaries and evaluation metrics. *Proceedings of the Text Analysis Conference*.

John M Conroy, Sashka T Davis, Jeff Kubina, Yi-Kai Liu, Dianne P Oleary, and Judith D Schlesinger. 2013. Multilingual summarization: Dimensionality reduction and a step towards optimal term coverage. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 55–63.

Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. 2012. OCCAMS - an optimal combinatorial covering algorithm for multi-document summarization. In Jilles Vreeken, Charles Ling, Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb, and Xindong Wu, editors, *ICDM Workshops*, pages 454–463. IEEE Computer Society.

Susan T. Dumais. 1994. Latent semantic indexing (LSI): TREC-3 report. In *TREC*, pages 105–115.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

George Giannakopoulos, Vangelis Karkaletsis, George A. Vouros, and Panagiotis Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. *TSLP*, 5(3).

George Giannakopoulos, George A. Vouros, and Vangelis Karkaletsis. 2010. Mudos-ng: Multi-document summaries using n-gram graphs (tech report). *CoRR*, abs/1012.2042.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Rada Mihalcea. 2005. Language independent extractive summarization. In *Proceedings of ACL 2005*, Ann Arbor, MI, USA.

Gerard Salton. 1991. The smart information retrieval system after 30 years - panel. In Abraham Bookstein, Yves Chiaramella, Gerard Salton, and Vijay V. Raghavan, editors, *SIGIR*, pages 356–358. ACM.

The Unicode Consortium. 2014. Unicode standard annex 29 : unicode text segmentation. `http://www.unicode.org/reports/tr29`.