Unsupervised Methods of Topical Text Segmentation for Polish

Dominik Flejter
University of Economics
al. Niepodległości 10
Poznań, Poland
D.Flejter@kie.ae.poznan.pl

Karol Wieloch
University of Economics
al. Niepodległości 10
Poznań, Poland
K.Wieloch@kie.ae.poznan.pl

Witold Abramowicz
University of Economics
al. Niepodległości 10
Poznań, Poland
W.Abramowicz@kie.ae.poznan.pl

Abstract

This paper describes a study on performance of existing unsupervised algorithms of text documents topical segmentation when applied to Polish plain text documents. For performance measurement five existing topical segmentation algorithms were selected, three different Polish test collections were created and seven approaches to text pre-processing were implemented. Based on quantitative results (P_k and WindowDiff metrics) use of specific algorithm was recommended and impact of pre-processing strategies was assessed. Thanks to use of standardized metrics and application of previously described methodology for test collection development, comparative results for Polish and English were also obtained.

1 Introduction

Rapid development of Internet-based information services is marked by a proliferation of information available on-line. Even if the Web shifts towards multimedia content and structured documents, contents of the Web sources remains predominantly textual and poorly structured; an important fraction of this flood of plain text documents consists in multi-topical documents. This abundance of complex but plain text or just visually structured documents (as is in case of most HTML files) creates a strong need for intelligent text processing including robust and efficient information extraction and retrieval.

One way of increasing efficiency of typical text processing tasks consists in processing separately text segments instead of whole documents. While different text segmentation strategies can be applied including splitting text into segments of equal length, using moving windows of constant size or discourse units (Reynar, 1998), division into topical segment is intuitively more justified. This approach is applicable in several IR and NLP areas including document indexing, automatic summarization and question answering (Choi, 2002).

For Information Extraction domain, two main application of topical document segmentation are related to documents pre-processing and supporting some basic tasks widely used by IE tools. Topical segmentation applied to individual documents as well as documents streams (e.g. dialogues of radio broadcast transcripts) is an initial step for further IE processing (Manning, 1998); when combined with segments labelling, classification or clustering (Allan et al., 1998) it allows to pre-select ranges of text to be mined for mentions of events, entities and relations relevant to users needs and available IE resources. This limits significantly size of text to be processed by IE methods (Hearst and Plaunt, 1993) and thus influences significantly overall IE performance.

On the other hand, many basic tasks required by IE domain (both related to IE resources creation and document (pre-)processing) make use of sections rather than whole documents. In these tasks including language modelling (esp. gathering of co-occurrences statistics for trigger-based language models), anaphora resolution, word sense disambiguation and coreference detection, definition of proper context is crucial. To this extend entities

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 51–58,
Prague, June 2007.

discovered by topical segmentation are more reliable than document, paragraph or sentence contexts as they help avoid usage of unrelated parts of documents as well as minimize sparse data problems (Choi, 2002).

1.1 Problem Statement

In this study we adopted definition proposed in (Flejter, 2006) stating that "*linear topical segmentation of text consists in extracting coherent blocks of text such that: 1) any block focuses on exactly one topic 2) any pair of consecutive blocks have different main topics*." We also accepted that two segments should be defined as having different or same topics whenever they are judged so by people.

Depending on the used document collection and user needs, text segmentation objective is to find topical segments in individual multi-topical text documents or to discover boundaries between consecutive stories or news items (e.g. in radio broadcast transcriptions or text news streams).

Approaches to the problem of text segmentation can be divided according to a number of dimensions (Flejter, 2006) including cognitive approach (in optimistic approach text structure intended by author is cognitively accessible, in pessimistic approach it is not), hierarchical or linear segmentation (do we segment text into some levels of embedded segments?), completeness of segmentation (does the segmentation need to cover the whole text?), disjointness of segments (can two segments have any common text ranges?), fuzziness of segments boundaries (how fuzzy is the actual boundary location?), global or local view of topics (is there any global, document-independent list of topics?).

In this study we investigate linear, complete, disjoint segments with binary boundaries and local view of topics. Selected algorithms focus on text segmentation without considering labelling or clustering of discovered segments.

1.2 Our Contributions

The contributions of our research described in this paper are twofold. Firstly, we developed three Polish test collections for text segmentation task (see: Section 3.1). Secondly, we performed an extensive study of performance of most popular segmentation algorithms (see: Section 3.2) and pre-processing strategies (see: Section 3.3) when applied to Polish documents; a total of 42 scenarios including different algorithms, pre-processing strategies and test collections were evaluated (see: Section 4).

2 Approaches to Topical Segmentation

As in case of most NLP tasks, a number of different linguistic theories influenced topic segmentation resulting in a variety of approaches applied. This section gives a short presentation of major theories underlying topical segmentation and an overview of most popular segmentation algorithms with emphasis on those evaluated in our experiment.

2.1 Theoretical Foundations

Out of linguistic approaches cohesion theory of Halliday and Hassan had the strongest impact on topical text segmentation. It analyzes several mechanisms of documents internal cohesion including references, substitution, ellipsis, conjunction (logical relations) and lexical cohesion (reuse of the same words to address the same object or objects of the same class as well as use of terms which are more general or semantically related in systematic or non-systematic way). Other relevant linguistic theories include Grosz and Sinder's discourse segmentation theory, Rhetorical Structure Theory and taxonomy of text structures proposed Skorochod'ko (Reynar, 1998).

Out of empirical statistical rules some authors make use of heuristics resulting form Heaps' law for new-words-based topical segment boundary detection. However, the most important theoretical foundations in quantitative methods are related to strong probabilistic frameworks including Hidden Markov Models (Mulbregt et al., 1998) and Maximal Entropy Theory (Beeferman et al., 1997).

2.2 Basic Methods

The most simple but also the most frequently used methods of topical text segmentation do not require training (thus they are domain-independent) nor make use of any complex linguistic resources or utilities. Apart from methods based on new vocabulary analysis, this category of algorithms applies widely the simplest form of lexical cohesion i.e. reiterations of the same word.

The first classical text segmentation algorithm of this type is TextTiling described in (Hearst and Plaunt, 1993). Its analysis unit consists of pseudo-sentences corresponding to series of consecutive words (typically 20 words). After the whole text is divided into pseudo-sentences, a window of 12 pseudo-sentences is slid over the text (with one pseudo-sentence step). At any position the window is decomposed into two six-pseudo-sentences blocks and their similarity is calculated by means of cosine measure. Measurements for all consecutive window positions (understood as positions of centre of the window) form lexical cohesion curve local minima of which correspond to segments boundaries. The original algorithm was further enhanced in several ways including use of words similarity measurement based on co-occurrences (Kaufmann, 1999).

Another group of basic algorithms makes use of technique of DotPlotting, originally proposed by Raynar in (Reynar, 1994). In this approach 2D chart is used for lexical cohesion analysis with both axes corresponding to positions (in words) in the text; on the chart points are drawn at coordinates (x, y) and (y, x) iff words at positions x and y are equal. In this settings coherent text segments correspond visually to squares with high density of points. DotPlotting image is than segmented using one of two strategies: minimization of points density at the boundaries (minimization of external incoherence) or maximization of density of segments (maximization of internal coherence) (Reynar, 1998). The original DotPlotting algorithm requires to explicitly provide expected number of segments as input.

Improved version of DotPlotting algorithm called C99 (Choi, 2000) uses DotPlotting chart for visualization of similarity measurements at consecutive point of the text (thus resulting in point with different levels of intensity) instead of words co-occurences. Afterwards, mask-based ranking technique is used for image enhancement. For actual segmentation, dynamic programming technique similar to DotPlotting maximization algorithm is used; an optional automatic termination strategy is also implemented thus allowing the algorithm to assess number of boundaries. In further work of the same and other authors several enhancements of C99 algorithm were proposed.

2.3 Methods Requiring External Resources

Still not requiring training and domain independent, some methods make use of linguistic resources more sophisticated than stop-list. Two classes of such solution described in existing work are solutions using lexical chains (Morris and Hirst, 1991; Min-Yen Kan, 1998) (which require to use some thesaurus) and based on spreading activation (Kozima, 1993) (which depend on weights-based semantic network constructed from thesaurus). In both cases the effort put in algorithm enactment is quite high; however in principle no additional resources need to be developed for new texts (even from different domains).

2.4 Methods Requiring Training

Last group of methods includes supervised methods with generally strong mathematical foundations. They perform very well; however they require training that possibly needs to be repeated when new domain needs to be addressed. The methods in this group use probabilistic frameworks including maximal entropy (Beeferman et al., 1997), Hidden Markov Models (Mulbregt et al., 1998) and Probabilistic Latent Semantic Analysis (Blei and Moreno, 2001).

3 Experimental Setup

Segmentation algorithms performance was evaluated for 42 scenarios corresponding to different algorithms, pre-processing strategies and test collections. For quantitative analysis and comparability with previous and future research results two standard segmentation metrics were applied in all scenarios.

3.1 Test Collections

For performance measurement three test collections corresponding to different types of segmentation tasks were developed: artificial documents collection (AC), stream collection (SC) and individual documents collection (DC). AC and SC were constructed based on 936 issues of EuroPAP (European information service of Polish Press Agency) plain-text e-mail newsletter (EuroPAP, 2005) collected from November 2001 to May 2005. Typical issue of EuroPAP newsletter contained about

25 complete news articles and a number of short (containing at most several sentences) news items. DC was constructed based on articles retrieved form Wikipedia corresponding to ten most populous Polish cities (Wikipedia, 2007) covering typically several topics (e.g. geography, culture, transportation).

For AC creation we followed precisely the method applied for English in (Choi, 2000). Each artificial document was created as a concatenation of random number (n) of first sentences from ten news articles randomly selected from a total of 24927 news items in EuroPAP corpus. Four subcollections were created depending on allowed range of n as listed in Table 1. Any two selected articles were assumed to cover two different topics; thus reference segmentation boundaries corresponded to points of concatenations.

AC	AC1	AC2	AC3	AC4
n	3-11	3-5	6-8	9-11
documents count	400	100	100	100

Table 1: Artificial collection subcollections

For SC creation newsletter messages from EuroPAP were used as text streams (936 messages). The reference segmentation was created using original article boundaries present in EuroPAP mail messages (almost 30000 segments were marked).

For individual documents collection development text content was extracted from Wikipedia documents, all headings were removed and all list items (LI tags) with no terminal punctuation sign were added a dot. Manual tagging by two authors of this paper was performed. The instructions were to put segment boundaries in the places of potential section titles. Obtained percent agreement of 0.988 and κ coefficient (Carletta, 1996) of 0.975 suggest high convergence of both annotations. Further, in places where the two annotators opinions differed (one marked boundary and the other did not), negotiation-based approach (Flejter, 2006) was applied in order to develop reference segmentation.

3.2 Selected algorithms

In our experiment we used Choi's publicly available implementation of several text segmentation algorithms not requiring training (with several adaptation concerning pre-processing stages). Specifically we used Choi's implementation of TextTiling algorithm (TT), C99 algorithm for both known ($C99_l$) and unknown ($C99$) number of boundaries as well as DotPlotting maximization (DP) and minimization (DP_{min}) algorithms.

	Sentences		Tokens	
	avg	std	avg	std
AC	6.8	1.7	122.6	33.4
SC	15.0	3.8	267.6	64.0
DC	28.5	9.9	300.0	110.2

Table 2: The average length of reference segments

Algorithms not requiring to provide number of segments as input (TT, $C99$) were evaluated on all test collections; performance of the remaining algorithms ($C99_l$, DP, DP_{min}) was measured only for AC.

3.3 Pre-processing variants

We decided to prepare seven variants of the test collections (see Table 3). The motivation for the first group (variants: P1, P2, P3, P4) was to be as close to Choi's methodology as possible. That's why we used simple pre-processing techniques like lemmatization and stop-lists. The remaining variants (P5, P6, P7) were chosen arbitrarily to check how additional morphological information will influence the performance of the main segmentation algorithms in case of Polish language.

The pre-processing stage included two steps. Initially, documents were split into sentences and word tokens (punctuation signs were removed) by means of tokenizer and sentence boundary recognizer of SProUT — a shallow text processing system tailored to processing Polish language (Piskorski et al., 2004). Afterwards the generated token stream was normalized; for this task SProUT's interface for a dictionary based Polish morphological analyzer — Morfeusz (Woliński, 2007) was used. This allowed us to use variety of morphological information (including STEM, POS, NUMBER, TENSE). The drawback of such an approach was that tokens not present in Morfeusz's dictionary were not stemmed (accounting for 12.8% of all tokens or 31.7% of unique tokens; note that Morfeusz input

Id	Variant	Description
P1	I	no changes (tokens remain inflected)
P2	L	lemmatized tokens
P3	LSL	L − words in the lemmatized stop-list
P4	ISI	I − without words in the inflected stop-list
P5	L-VT	L + verbs tagged with POS and TENSE
P6	L-VT-N-A	L-VT + nouns and adjectives tagged with POS
P7	L-VT-NN-AN	L-VT-N-A + nouns and adjectives tagged with NUMBER

Table 3: pre-processing variants

contained all the tokens including numbers, hyperlinks, dates, etc.). Another problematic issue was ambiguity of morphological analysis (1.4 interpretation on average); we addressed this issue by using the following order of preference: 1) verbs, 2) nouns, 3) adjectives, 4) other word classes.

The stop-lists (inflected and lemmatized versions contained 616 and 350 tokens respectively) were prepared manually by analysing frequency lists of previously used text corpus.

3.4 Evaluation metrics

A number of different measurement methods were applied to topical texts segmentation including recall-precision pair (Hearst and Plaunt, 1993; Passonneau and Litman, 1997), edit distance (Ponte and Croft, 1997), P_μ (Beeferman et al., 1997), P_k (Beeferman et al., 1999) and *WindowDiff* (Pevzner and Hearst, 2002).

P_k is simplified version of probabilistic measure P_μ based on assumption that any two consecutive boundaries are at distance of k sentences (k being parameter normally set to half of length of average segment in reference segmentation). After some simplifications P_k is defined by the following formula (Flejter, 2006):

$$P_k(r,h) = 1 - \frac{1}{n-k}\sum_{i=1}^{n-k}(|\delta_r(i,k) - \delta_h(i,k)|)$$

where $\delta_X(i,k)$ equals to one if ith and $(i+k)$th sentences are in the same segment of segmentation X, otherwise it is equal to zero; $X = r$ corresponds to reference segmentation and $X = h$ corresponds to hypothetical (algorithm-generated) segmentation.

In most publication instead of performance measurement using P_k, probabilistic error metric ($P = 1 - P_k(r,h)$) is applied. For easier comparison with previous results we calculated this measure for tested evaluation scenarios.

Based on a profound analysis of P_k and probabilistic metric drawbacks more recently WindowDiff error measure was proposed based on counting number of boundaries within window of size of k sentences sliding parallelly over both hypothetical and reference segmentations. WindowDiff can be calculated by the following formula:

$$W_k(r,h) = \frac{1}{N-k}\sum_{i=1}^{N-k}(|b_r(i,k) - b_h(i,k)| > 0)$$

with $b_X(i,k)$ corresponding to the number of boundaries between positions i and $i+k$ in segmentation X.

Both probabilistic error and WindowDiff measure the segmentation error; therefore, the lower is their value, the better is segmentation result.

4 Experimental Results

Calculated values of P and WindowDiff (WD) measures were used to compare performance of different algorithms, collections and pre-processing strategies. If not stated otherwise, results displayed in this Section correspond to P1 variant (no pre-processing) of test collections.

	$C99$		TT	
	P	WD	P	WD
AC	0.365	0.355	0.527	0.638
AC1	0.360	0.350	0.539	0.639
AC2	0.387	0.360	0.435	0.436
AC3	0.370	0.360	0.549	0.650
AC4	0.359	0.364	0.551	0.821
SC	0.381	0.390	0.562	0.932
DC	0.429	0.477	0.554	0.877

Table 4: Comparison of methods not requiring number of segments as input

Comparative results of both algorithms not requiring to provide expected number of segments as input (i.e. $C99$ and TT) are listed in Table 4. $C99$

performs much better than TextTiling on all test collections with extremely high WindowDiff value for TextTiling (especially for longer texts). Both algorithms perform better on artificial than on actual documents; for $C99$ drop in performance between stream and cities documents is also visible.

	$C99_l$		DP		DP_{min}	
	P	WD	P	WD	P	WD
AC	0.339	0.332	0.353	0.338	0.462	0.485
AC1	0.342	0.336	0.368	0.353	0.460	0.483
AC2	0.324	0.304	0.343	0.318	0.455	0.483
AC3	0.342	0.337	0.347	0.332	0.464	0.487
AC4	0.338	0.341	0.310	0.300	0.475	0.495

Table 5: Comparison of methods requiring number of segments as input

Probabilistic error and WindowDiff results for algorithms requiring expected number of segments as input ($C99_l$, DP, DP_{min}) are listed in Table 5. $C99_l$ performs slightly better than DotPlotting with maximization strategy (DP) and the performance of DotPlotting applying minimization strategy (DP_{min}) is visibly lower. Results do not differ significantly between AC subcollections suggesting that length of document has minor impact on performance.

As $C99$ was developed both in version requiring and not requiring to specify the expected number of segments, impact of this information on algorithm performance was analyzed. As expected $C99_l$ outperforms $C99$; in our experiments additional information on segments count lowered the error rates by 4–16% (see: Table 6).

As previous research on English text segmentation (Choi, 2000) was led for the same artificial collection creation methodology (see section 3.1), algorithm implementations and probabilistic error metric, comparison of algorithms performance for Polish and English was possible. The results of such comparison are gathered in Table 7; for English results were taken from Choi's paper and for Polish P3 variant of our artificial collection corresponding to Choi's pre-processing approach was used. Comparative analysis shows that all algorithms (except for TT which is highly inefficient for both Polish and English) perform significantly worse for Polish. Our hypothesis is that it can be attributed both to lower performance of pre-processing tools for Polish and usage of domain specific corpus as opposed to balanced Brown corpus used by Choi.

	$C99$		$C99_l$		$\Delta\%$	
	P	WD	P	WD	P	WD
AC	0.365	0.355	0.339	0.332	7%	6%
AC1	0.360	0.350	0.342	0.336	5%	4%
AC2	0.387	0.360	0.324	0.304	16%	16%
AC3	0.370	0.360	0.342	0.337	8%	6%
AC4	0.359	0.364	0.338	0.341	6%	6%

Table 6: Impact of segments count provided as input

		3-11	3-5	6-8	9-11
$C99(P3)$	PL	0.32	0.34	0.31	0.29
	EN	0.13	0.18	0.10	0.10
$C99_l(P3)$	PL	0.30	0.27	0.29	0.28
	EN	0.12	0.12	0.09	0.09
$DP(P3)$	PL	0.32	0.28	0.26	0.24
	EN	0.22	0.21	0.18	0.16
$DP_{min}(P3)$	PL	0.46	0.46	0.46	0.47
	EN	n/a	0.34	0.37	0.37
$TT(P3)$	PL	0.50	0.39	0.49	0.54
	EN	0.54	0.45	0.52	0.53

Table 7: Polish versus English results

	$AC + C99_l$		$SC + C99$		$DC + C99$	
	P	WD	P	WD	P	WD
P1	0.365	0.355	0.381	0.390	0.429	0.477
P2	0.322	0.315	0.356	**0.367**	0.433	0.477
P3	**0.319**	**0.311**	**0.354**	0.368	0.452	0.497
P4	0.342	0.334	0.381	0.391	0.425	0.473
P5	0.362	0.318	0.356	0.368	0.435	0.480
P6	0.416	0.403	0.413	0.419	**0.406**	**0.430**
P7	0.416	0.403	0.413	0.419	**0.406**	**0.430**
$\Delta\%$	13%	12%	7%	6%	5%	10%

Table 8: Impact of different pre-processing variants

Final part of our analysis of quantitative results focused on impact of different pre-processing strategies. For each of three test collections we analyzed the impact of pre-processing strategies in case of best performing algorithm.

As the results from Table 8 show, for artificial and stream collections the most promising strategy is to use P3 (LSL) variant of pre-processing which decreased the error metric by up to 13% as compared with P1 (no pre-processing) variant. Interestingly, the same approach applied to individual documents collection actually increased values of error metrics; in this case significant decrease of error rates was possible by use of the most complex P6 and P7 strategies. Reasons for this difference may include: a) disproportion in number of unrecognized tokens (22.8% for DC vs. 12.75% for AC/SC), b) different structure of DC reference segments (higher number of shorter sentences, see: Table 2), c) standard deviation of segment length in DC much higher than in AC/DC (35% of average length in case of DC vs. 25% in case of both AC and SC). We leave analysis of this factors' impact for future work.

Our analysis also shows that adding NUMBER tag for nouns and adjectives (P7) has no impact on algorithms performance as compared with P6.

5 Conclusions and Future Work

In this paper we analyzed performance of several topical text segmentation algorithms for Polish with several pre-processing strategies on three different test collections. Our research demonstrate that similarly to English $C99$ (and its variant with expected segments count input $C99_l$) is the best performing segmentation algorithm and we recommend that it be applied to text segmentation for Polish. Based on our research we also suggest that lemmatization and stop-list words removal (P3 variant) be used for further improvement of performance. However, our research revealed that the performance of almost all algorithms (including $C99$) is significantly worse for Polish than for English and remains unsatisfactory.

Therefore our further research direction will be to focus on improvements at the pre-processing stages of text segmentation (including enhancements in text division into sentences, lemmatization of proper names, and filtering of unrecognized tokens with low document-based frequency) as well as on analysis of performance of more recent algorithms both requiring and not requiring linguistic resources. We would like also to evaluate text segmentation impact on performance of coreference resolution algorithm we are currently developing.

References

James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study. final report.

Doug Beeferman, Adam Berger, and John Lafferty. 1997. Text segmentation using exponential models. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 35–46. Association for Computational Linguistics, Somerset, New Jersey.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

David M. Blei and Pedro J. Moreno. 2001. Topic segmentation with an aspect hidden Markov model. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 343–348, New York, NY, USA. ACM Press.

Jean Carletta. 1996. Assessing agreement on classification task: the kappa statistic. *Computational Linguistic*, 22:250–254.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 26–33, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Freddy Y. Y. Choi. 2002. *Content-based Text Navigation*. Ph.D. thesis, Department of Computer Science, University of Manchester.

EuroPAP. 2005. Serwis o Unii Europejskiej, http://euro.pap.com.pl/ (2001-2005).

Dominik Flejter. 2006. Automatic topical segmentation of documents in text collections (in polish). Master's thesis, Poznan University of Economics, June.

Marti A. Hearst and Christian Plaunt. 1993. Subtopic structuring for full-length document access. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–68, New York, NY, USA. ACM Press.

Stefan Kaufmann. 1999. Cohesion and collocation: Using context vectors in text segmentation. In *Proceedings of the 37th annual meeting of the Association for*

Computational Linguistics on Computational Linguistics, pages 591–595, Morristown, NJ, USA. Association for Computational Linguistics.

Hideki Kozima. 1993. Text segmentation based on similarity between words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 286–288, Morristown, NJ, USA. Association for Computational Linguistics.

C. Manning. 1998. Rethinking text segmentation models: An information extraction case study. Technical report, University of Sydney.

Kathleen R. McKeown Min-Yen Kan, Judith L. Klavans. 1998. Linear segmentation and segment significance. In *Proceedings of 6th International Workshop of Very Large Corpora (WVLC-6)*, pages 197–205.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

P. Mulbregt, I. van, L. Gillick, S. Lowe, and J. Yamron. 1998. Text segmentation and topic tracking on broadcast news via a hidden markov model approach. In *Proceedings of the ICSLP'98, volume 6*, pages 2519–2522.

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Comput. Linguist.*, 23(1):103–139.

Lev Pevzner and Marti A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28(1):19–36.

Jakub Piskorski, Peter Homola, Malgorzata Marciniak, Agnieszka Mykowiecka, Adam Przepiórkowski, and Marcin Wolinski. 2004. Information extraction for Polish using the SProUT platform. In *Intelligent Information Processing and Web Mining, Proceedings of the International Intelligent Information Systems: IIPWM'04 Conference*, Advances in Soft Computing, pages 227–236. Springer.

Jay M. Ponte and W. Bruce Croft. 1997. Text segmentation by topic. In *ECDL '97: Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 113–125, London, UK. Springer-Verlag.

Jeffrey C. Reynar. 1994. An automatic method of finding topic boundaries. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 331–333, Morristown, NJ, USA. Association for Computational Linguistics.

Jeffrey C. Reynar. 1998. *Topic segmentation: algorithms and applications. A dissertation in Computer and Information Science*. Ph.D. thesis, University of Pennsylvania.

Wikipedia. 2007. Miasta w Polsce wedlug liczby ludnosci, Accessed on 20.03.2007.

Marcin Woliński. 2007. Analizator morfologiczny Morfeusz SIAT. On-line. Accessed on: 30.01.2007.