7

Segmentation of Greek Texts by Dynamic Programming

Pavlina Fragkou[1], Athanassios Kehagias[2] and Vassilios Petridis[1]
[1]*Department of Electrical and Computer Engineering,*
[2]*Department of Math., Phys., and Computer Sciences,*
Faculty of Engineering, Aristotle University of Thessaloniki,
Greece

1. Introduction

In this paper we present an approach for the segmentation of concatenated texts. The text segmentation problem can be stated as follows: given a text which consists of several parts (each part dealing with a different subject) it is required to find the boundaries between the parts. In other words, the goal is to divide a text into homogeneous segments so that each segment deals with a particular subject while contiguous segments deal with different subjects. In this manner, documents relevant to a query can be retrieved from a large database of unformatted (or loosely formatted) text. The problem appears often in information retrieval and text processing.

Our approach combines elements from several previously published text segmentation algorithms and achieves a significant improvement in segmentation accuracy by following a supervised approach. More specifically, we perform linear segmentation of concatenated texts by minimizing a segmentation cost which consists of two parts: (a) within-segment word similarity (expressed in terms of dotplot density) and (b) prior information about segment length. The minimization is effected by dynamic programming, which guarantees that the globally optimal segmentation is obtained. We are concerned with linear text segmentation, which should be distinguished from hierarchical text segmentation (Yaari, 1997; Yaari, 1999); the latter attempts to find a tree-like structure in the text segments, while linear segmentation is based on the assumption that text has a linear structure thus segments appear in sequential "flat" order. Let us note that hierarchical segmentation is perhaps more appropriate for discourse segmentation because it creates a hierarchy of all topics discussed. Every sub-topic is appropriately related to the topic with which is related to in a deeper level placed in a form of "leaf".

Our method has successfully applied to Greek texts proving to be very innovating and promising. Results regarding segmentation of English texts can be found in (Kehagias et al., 2004(a); Kehagias et al., 2004(b)). The remainder of the paper is organized as follows: in Section 2 we present research approaches on the area of text segmentation, in Section 3 we introduce our algorithm, in Section 4 we present experiments to evaluate the algorithm. Finally, in Section 5 we discuss our results.

2. Related work

Text segmentation approaches are based in the theory of Halliday and Hasan (Halliday & Hasan, 1976) according to which, each text is described by two complementing elements: *cohesion* and *coherence*. *Cohesion* is described as the quality property of a text and is detected by the simultaneous appearance of semantically similar words. Cohesion is present when an element in the text is best interpreted in light of a previously (or rarely a subsequent) element within the text. *Coherence* on the other hand holds between two tokens in the text which are either of the same type or are semantically related in a particular way (such as a word or group of words having a clearly definable relationship with a previously used word i.e. belonging to the same theme or topic). According to Halliday and Hasan semantic coherence and cohesion are identified by the following five semantic relations: (1) repetition with similarity, (2) repetition without similarity (3) repetition through reference to a higher category in which the aforementioned word entity belongs to (4) systematic semantic relationship (5) non- systematic semantic relationship. In the same spirit, Raskin and Weiser (Raskin & Weiser, 1987) defined as a criterion for cohesion and coherence word repetition and comparative apposition, where the first focus on word repetition or synonyms of them and the latter on words that present the tendency to co-occur within a document.

In this paper, the focus is stressed towards (concatenated) text segmentation, which is often distinguished from discourse segmentation. The goal of discourse segmentation is to split a single large text into its constituent parts (e.g. to segment an article into sections); this problem is addressed, for instance, in (Hearst, 1994; Hearst & Plaunt, 1993; Heinonen, 1998; Yaari, 1997; Yaari, 1999). On the other hand, the goal of (concatenated) text segmentation is to split a stream of independent, concatenated texts (e.g. to segment a transcript of news into separate stories); this problem is addressed, for example, in (Beeferman et al., 1999; Choi, 2000; Choi et al., 2001; Ponte & Croft, 1997; Reynar, 1994; Reynar & Ratnaparkhi, 1997; Utiyama & Isahara, 2001). The two problems are similar but not identical; our algorithm could conceivably be applied to discourse segmentation, but our main interest is in concatenated text segmentation and all the experiments we present here fall into this category.

Generally speaking, text segmentation is a two step procedure. The first step involves the calculation of segment *homogeneity* while the second the identification of segment boundaries. The calculation of segment *homogeneity* (or alternatively *heterogeneity*) performed by methods appearing in the literature presents a strong variation. On the one hand, a family of methods makes use of linguistic criteria such as cue phrases, punctuation marks, prosodic features, reference, syntax and lexical attraction (Beeferman et al. (1997), Hirschberg & Litman (1993), Passoneau & Litman (1993)). On the other hand the second family, following Halliday and Hasan's theory (Halliday & Hasan (1976)), utilizes statistical similarity measures such as word co-occurrence. Roughly speaking, two parts of the text are considered similar if they have many words in common. This is a popular approach, according to which parts of a text having similar vocabulary are likely to belong to a coherent topic segment. For example the linear discourse segmentation algorithm proposed by Morris and Hirst (Morris & Hirst (1991)) is based on *lexical cohesion relations* determined by use of Roget's thesaurus (Roget (1977)). In the same direction Kozima's algorithm (Kozima (1993), Kozima & Furugori (1993)) computes the semantic similarity between words using a semantic network constructed from a subset of the Longman Dictionary of Contemporary English. Local minima of the similarity scores correspond to the positions of

topic boundaries in the text. Other authors have used fairly sophisticated word co-occurrence statistics such as LSA, LCA, ranking etc. Choi, 2000; Choi et al., 2001; Hearst, 1994; Hearst & Plaunt, 1993; Utiyama & Isahara, 2001).

The identification of segment boundaries usually requires the minimization of a segmentation cost function. An efficient way to perform this is by the use of techniques such as dynamic programming. This is due to the fact that dynamic programming is based on the intuition that a longer problem can be solved by properly combining the solution to various sub-problems. For example, consider the sequence or "path" of transformed words that comprise the minimum edit distance between the strings "intention" and "exention". Imagine one string (perhaps it is exention) that is in this optimal path (whatever it is). The intuition of dynamic programming is that if exention is in the optimal operation list, then the optimal sequence must also include the optional path from intention to exention. This is because, if there were a shorter path from intention to exention then we could use it instead, resulting in the shortest path and the optimal sequence wouldn't be optimal, thus leading to contradiction. Another benefit of dynamic programming is that at every point of execution the optimal solution from the previously examined observations was calculated avoiding thus backtracking (Berteskas, 1987). This approach has been used in the past (in Heinonen, 1998; Ponte & Croft, 1997; Xiang & Hongyuan, 2003) and also, implicitly, in (Utiyama & Isahara, 2001). Other authors do not cast segmentation as a formal optimization problem; rather they construct a similarity matrix which they segment using divisive clustering, which can be considered as a form of approximate and local optimization (Choi, 2000; Choi et al., 2001; Reynar, 1994; Reynar & Ratnaparkhi, 1997; Yaari, 1997; Yaari, 1999).

As we have already mentioned, we formulate segmentation as the minimization of a segmentation cost which depends on within-segment homogeneity and deviation from expected segment length. We measure within-segment homogeneity by word co-occurrence by operating at the sentence level and consider two sentences to be similar if they have even a single word in common. We use a "global" similarity comparison, i.e. we evaluate the similarity between all parts of a text (for example between every pair of sentences that appear in the text, even if they are not adjacent to each other). This approach is used by several authors (Choi, 2000; Choi et al., 2001; Ponte & Croft, 1997; Reynar, 1994; Reynar & Ratnaparkhi, 1997; Xiang & Hongyuan, 2003), but it should be noted that "local" comparison (i.e. only between adjacent sentences) has also been used in the past (Hearst, 1994; Hearst & Plaunt, 1993; Heinonen, 1998). To penalize deviations from the expected segment length we use a "length-model"; this approach has been used in the past by several authors (Heinonen, 1998; Ponte & Croft, 1997). We find the globally minimal segmentation cost by dynamic programming.

Current approaches to text segmentation include an improvement of the dotplotting technique (Ye et al., 2005) introduced by Reynar (Reynar, 2004), an improvement of Latent Semantic Analysis for text segmentation (Bestgen, 2006), a model of text segmentation based on ideas from multilabel classification for segmenting sentences into tokens (McDonald et al., 2005) as well as a novel parameter-free unsupervised text segmentation method, which is formulated as (variational) Bayes estimation of an HMM from an input text stream (Koshinaka et al., 2005). Teo Yung Kiat' master thesis present an attempt to extend and improve our method (Kiat, 2005). Advances to topic segmentation (closely related to text segmentation) include methods performing topic segmentation method based on weighted lexical chains (Sitbon & Bellot, 2005), as well as a new informative similarity measure based on word co-occurrences (Dias & Alves, 2005).

It is worth mentioning that text segmentation is widely used in other closely related scientific areas such as speech segmentation i.e. to identify breaks and discourse boundaries by expert and/or naive listeners (Auran et al., 2005), spoken multiparty dialogue and tutorial dialogue segmentation (Olney & Cai, 2005; Hsueh et al., 2006). Text segmentation techniques are also applied to entity extraction and noun-phrase chunking (Ursu et al., 2005) as well as to semantic annotation of transcripts of television news broadcasts produced through automatic speech recognition (ASR) (Dowman et al., 2005). Text segmentation proves to be beneficial in a number of scientific areas such as corpus linguistics, discourse psychology and even education. This is due to the fact that text segmentation is based on topic change. Topic change or topic coherence is highly related to the vocabulary used by each author, the subconscious mechanism of language variation, the part of speech of words that he/she uses which may reveal positivity, sociability, complexity or negativity, self concern emphasis and implicitness. In psychological perspective, text segmentation may reveal if an author express its subject in question by following a coherence and progressive apposition of his arguments or it interrupts his argumentation by making references to less important or even non relevant subjects. Thus, text segmentation can be found useful in studies concerning topic and authorship attribution where topic change can highly be related to the vocabulary used by each author (Stamatatos et al., 2001). Finally, text segmentation can easily be applied as a preliminary step to text summarization.

3. Method and algorithm

3.1 Text representation

A text consists of words which are organized in sentences. We assume that sentence boundaries are correctly marked in the text. Hence we will assume from now on that the basic text unit is the sentence and that segment boundaries occur only at the end of sentences. Consider a text which contains T sentences and L distinct words (i.e. a vocabulary of size L). We define a T x T similarity matrix D as follows (s, t = 1, 2,…, T)

$$D_{s,t} = \begin{cases} 1, \textit{if sentences s and t have at least a common word and } s \neq t \\ 0, \textit{otherwise} \end{cases} \quad (1)$$

It is worth mentioning that, by the term "words" we mean any word used by the author of that segment but not its grammatical form. In our study we do not perform an in depth linguistic process i.e. grammatical parsing and co-reference resolution in order to discover the context under which each word appears or the sequence of appearance of words. Our research is based on the hypothesis that each segment corresponds to a different topic. The description of that topic tends to be performed by using a small number of characteristic words that belong to a limited size vocabulary. On the other hand, highly informative words tend to appear more that one times, thus, the importance of them is reinforced in the similarity matrix. Finally, it is worth mentioning that, none of the algorithms dealing with the same problem make use of grammatical items. An opposite approach would lead to a misleading comparison of obtained results. Additionally, it is our belief that, it is the choice of words that the authors use in order to express their topic than the grammatical property of those that it acts as a discriminative factor in the topic i.e. segment change identification. Lastly, we believe that in case where high informative combination of words i.e. n-grams appear in the segment, the fact that the information that they contain is represented not as a

whole but with their consisting words as individuals does not lead to "loss" of the information contained.

Hence, if $D_{s,t} = 1$ we assume that the s-th and t-th sentence are similar. Figure 1 provides the dotplot (Choi, 2000; Choi et al., 2001; Reynar, 1994; Reynar & Ratnaparkhi, 1997) of a D matrix corresponding to a 91-sentences text; black squares correspond to 1's and white squares to 0's. Consecutive groups of sentences which have many words in common appears as submatrices of D with many 1's; in Figure 1 they appear as high density squares. Candidate segments appear, for example, between sentences 11 and 18, 41 and 52 etc. Hence the dotplot gives a visual representation of the structure of the text.

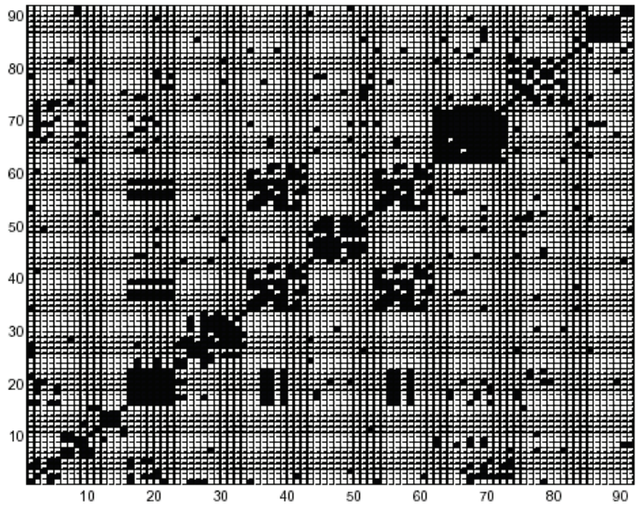


Figure 1: The similarity matrix D corresponding to a text containing 91 sentences, hence D is a 91 x 91 matrix. A black dot at position (s, t) indicates that the s-th and t-th sentence have at least one word in common

It is worth mentioning that, the total number of shared words is indirectly depicted in the dotplot similarity matrix. Sentences that have an important number of shared words lead to regions containing a lot of '1's. Sentence length is not considered here, as it would require the calculation of the total number of words belonging to each sentence, the number of common and non common words between sentences as well as sentence length normalization. Such approach is left for future research.

3.2 Segmentation cost

A segmentation is a partition of the set {1,2,…,T} into K subsets (i.e. segments) of the form $\{1,2,..,\ t_1\}$, $\{t_1+1,\ t_1+2,\ldots,\ t_2\}$, ..., $\{t_{K-1}+1,\ t_{K-1}+2,\ldots,T\}$(where K is a variable number and K ≤ T). A more economical description of the segmentation is given by a (variable length) vector $t = (t_0, t_1, \ldots, t_K)$, where, $t_0,\ t_1,\ \ldots,\ t_K$ are the segment boundaries which satisfy $0 = t_0 < t_1 < \ldots < t_K - 1 < t_K = T$.

We now introduce a "segmentation cost" function J(t): for every segmentation t, J(t) returns a real number; J(t) will be designed in such a way that it achieves small values when t designates high-density submatrices of D. We start with the function

$$J_0(t) = \frac{\sum_{s=t_{k-1}+1}^{t_k} \sum_{t=t_{k-1}+1}^{t_k} D_{s,t}}{\left(t_k - t_{k-1}\right)^r} \tag{2}$$

which can be interpreted as follows. The numerator is the total number of 1's contained into the D submatrix which corresponds to the k-th segment $\{t_{k-1}+1, t_{k-1}+2, \ldots, t_k\}$. When the parameter r=2, the denominator $\left(t_k - t_{k-1}\right)^r$ corresponds to the area of the sub-matrix and $J_0(t)$ is the "segment density". In the case r≠2, $J_0(t)$ corresponds to a "generalized density" which balances the degree of influence of the surface with regard to the "information" (i.e. the number of 1's) included in it. A "good" segmentation t is characterized by large values of $J_0(t)$, which indicate strong within-segment similarity.

In many cases some information will be available regarding the expected segment length; for instance we may use training data to estimate its mean value μ and standard deviation σ. We incorporate this information into a function:

$$J_1(t) = \sum_{k=1}^{K} \frac{\left(t_k - t_{k-1} - \mu\right)^2}{2 \bullet \sigma^2} \tag{3}$$

A "good" segmentation t is characterized by small values of $J_1(t)$, which indicate small deviation from the expected segment length (1).

Finally, we form J by a weighted combination of J_0 and J_1:

$$J(t; \mu, \sigma, r, \gamma) = \gamma \bullet J_1(t) - (1-\gamma) \bullet J_0(t) = \sum_{k=1}^{K} \left[\gamma \bullet \frac{\left(t_k - t_{k-1} - \mu\right)^2}{2 \bullet \sigma^2} \right] - (1-\gamma) \bullet \frac{\sum_{s=t_{k-1}+1}^{t_k} \sum_{t=t_{k-1}+1}^{t_k} Ds,t}{\left(t_k - t_{k-1}\right)^r} \tag{4}$$

where we stress the dependence of J on the parameters, μ, σ, r and γ.

3.3 Minimization by dynamic programming

A "good" segmentation vector t yields a small value of the corresponding $J(t; \mu, \sigma, r, \gamma)$ (i.e. segments with high density and small deviation from average segment length). The optimal segmentation $\hat{t}$ is the one which yields the global minimum of $J(t; \mu, \sigma, r, \gamma)$; note that $\hat{t}$ specifies not only the optimal positions of the segment boundaries $t_0, t_1, \ldots, t_K$ but also the optimal number of segments K; in other words, our algorithm automatically determines the optimal K.

---

[1] Many other functional forms can be used for $J_1(t)$; in Kehagias et al., 2004(a) and Kehagias et al., 2004(b), we have explored some alternatives but we have found that the form used here gives the best results.

Our J(t; μ, σ, r, γ) has an additive form which is well suited for the global minimization by dynamic programming. The following algorithm implements the basic dynamic programming idea (for a detailed justification the reader can consult (Bertsekas, 1987)).

Dynamic Programming for Text Segmentation

Input: The T x T similarity matrix D; the parameters μ , σ , r, γ :

Initialization

For t = 1, 2,…,T

q = 0

For s = 1,2, ..., t -1

$q = q + D_{s,t}$

$S_{s+1,t} = \frac{q}{(t-s)^r}$

End

End

Minimization

$C_0 = 0, Z_0 = 0$

For t = 1, 2,…,T

$C_t = \infty$

For s = 0, 1, ..., t -1

If

$$C_s + \gamma \bullet \frac{(t-s-\mu)^2}{2 \bullet \sigma^2} - (1-\gamma) \bullet S_{s+1,t} \le C_t$$

Then

$$C_t = C_s + \gamma \bullet \frac{(t-s-\mu)^2}{2 \bullet \sigma^2} - (1-\gamma) \bullet S_{s+1,t}$$

$$Z_t = s$$

EndIf

End

End

BackTracking

K = 0, $s_K = T$

While $Z_{s_K} > 0$

K = K + 1

$s_K = Z_{s_{K-1}}$

End

K = K + 1, $s_K = 0$, $\hat{t}_0 = 0$

For k = 1, 2,…,K

$\hat{t}_k = S_{K-k}$

End

Output: The optimal segmentation vector $\hat{t} = (\hat{t}_0, \hat{t}_1, \ldots, \hat{t}_K)$.

Upon completion of the minimization part of the algorithm we have computed the optimal segmentation cost for sentences 1 until T, i.e. for the entire text. The backtracking part first creates the sequence $s_0, s_1, .., s_K$ which are the optimal segment boundaries in reverse order and then reverses this sequence to produce the optimal $\hat{t} = (\hat{t}_0, \hat{t}_1, \ldots, \hat{t}_K)$. Note that K, the optimal number of segments is computed automatically.

4. Experiments - results

In this section we present the experiments we conducted to evaluate our algorithm. We evaluate the algorithm using the following three indices: *Precision, Recall* and Beeferman's P_k metric (Beeferman et al., 1999). *Precision* is defined as "the number of the estimated segment boundaries which are actual segment boundaries" divided by "the number of the estimated segment boundaries". *Recall* is defined as "the number of the estimated segment boundaries which are actual segment boundaries" divided by "the number of the true segment boundaries". It is worth mentioning that the F measure, which combines the results of Precision and Recall, is not used here, due to the fact that both Precision and Recall penalize equally segment boundaries that are "close" to the actual i.e. true boundaries with those that are less close to the true boundary. For that reason, Beeferman proposed an new metric P_k which measures segmentation inaccuracy; intuitively, P_k measures the proportion of "sentences which are wrongly predicted to belong to different segments (while actually they belong to the same segment)" or "sentences which are wrongly predicted to belong to the same segment (while actually they belong in different segments)" (for a precise definition of P_k see (Beeferman et al., 1999).

The variation of the P_k measure named WindowDiff index which was proposed by Pevzner and Hearst (Pevzer & Hearst, 2002) and remedies several problems of the P_k measure is not used in this paper due to the number of experiments conducted and the fact that already published results used for comparison are only reported in terms of P_k.

While several papers regarding the segmentation of English texts have appeared in the literature, we are not aware of any similar work regarding Greek texts. Furthermore, because Greek is a highly inflected language (much more than English) the segmentation problem is harder for Greek, as will be explained in the following. Hence some enhancements to the basic segmentation algorithm are required.

In the sequel we present experiments which use a Greek text collection compiled from Stamatatos'corpus 2(Stamatatos et al., 2001) comprising of text downloaded from the website http://tovima.dolnet.gr of the newspaper entitled 'To Vima'. This newspaper contains articles belonging to one of the following categories: 1) Editorial, diaries, reportage, politics, international affairs, sport reviews 2) cultural supplement 3) Review magazine 4) Business, finance 5) Personal Finance 6) Issue of the week 7) Book review supplement 8) Art review supplement 9) Travel supplement. Stamatatos et al. (Stamatatos et al., 2001)

[2] The authors would like to thank professor E. Stamatatos for providing us the corpus of Greek articles.

constructed a corpus collecting texts from supplement no. 2) which includes essays on science, culture, history etc. Stamatatos et al. selected 10 authors and used 30 texts per author. They didn't perform any manual text preprocessing or text sampling; however, they removed all the unnecessary heading irrelevant to the text itself. In order to minimize the potential change of the personal style of an author over time, they chose to download texts taken from the issues published from 1997 till early 1999. The thematic areas of each author are shown in Table 1.

Due to the nature of the newspaper supplement, texts included in, undergo some low-level post editing -as opposed to editorial or reportage articles, which are subject to a stricter editing- so that they conform to the overall style of the newspaper. Therefore, the style of the specific authors is more personal and independent of outer influences. An example of those documents is listed in Appendix B.

Author	**Thematic Area**
Alachiotis	Biology
Babiniotis	Linguistics
Dertilis	History, Society
Kiosse	Archeology
Liakos	History, Society
Maronitis	Culture, Society
Ploritis	Culture, History
Tassios	Technology, Society
Tsukalas	International Affairs
Vokos	Philosophy

Table 1. List of Authors and their Thematic Areas in the Stamatatos's collection of Greek texts.

We created several texts, each consisting of segments by various authors. Each author is characterized by her/his vocabulary hence our goal is to segment the text into the parts written by the various authors. Before creating the actual texts, some preprocessing (performed in a totally automatic manner) of the Stamatatos collection was necessary. Because Greek is a heavily inflected language, a word may appear in many different forms. Then, if one considers each inflected form as a separate element of the vocabulary, the result is a larger vocabulary, which considerably complicates the segmentation problem. To address this issue, we must identify various inflected forms as belonging to the same word; but for Greek this cannot be done using a simple approach such as stemming. Instead, we used the POS tagger developed by Orphanos et al. (see Orphanos & Christodoulakis, 1999; Orphanos & Tsalidis,1999) and the Appendix A, 3) to substitute each word by a "canonical", lemmatized form. More specifically, at the first stage, punctuation marks and numbers were removed as well as all words that aren't either nouns, verbs, adjectives or adverbs (the stop list used here is very similar to the one used for English texts). After that, every remaining word in the text was substituted by its lemma, determined by the tagger. In case the tagger could not find the lemma of a particular word (usually this happened because the word was

[3] The authors would like to thank professor G. Orphanos for kindly letting us use the POS Tagger.

not contained in the tagger Lexicon) no substitution was made and the word was kept in the form appearing in the text. We also kept the information regarding sentence ends.
We present two groups of experiments, which differ in the length of segments created and the number of authors used for the creation of the texts to segment.

4.1 Experiment group 1

The collection of texts used for the first group of experiments consists of 6 datasets: Set0,..., Set5. Each of those datasets differ in the number of authors used for the generation of the texts to segment and consequently in the number of texts used from the entire collection, as listed in Table 2.
For each of the above datasets, we constructed four subsets, which differ in the number of the sentences appearing in each segment. Let L_{min} and L_{max} be the smallest and largest number of sentences which a segment may contain. We have used four different (L_{min}, L_{max}) pairs: (3,11), (3,5), (6,8) and (9,11). Hence Set0 contains 4 subsets: Set01, Set02, Set03 and similarly for Set1, Set2, ..., Set5. The datasets Set*1 are the ones with (L_{min}, L_{max}) = (3,11), the datasets Set*2 are the ones with (L_{min}, L_{max}) =(3,5), and so on. Let also $\{X_1,...,X_n\}$ be the authors contributing to the generation of the dataset. We generated the texts in the dataset by the following procedure.
Each text is the concatenation of ten segments. For each segment we do the following.

1. We select randomly an author from $\{X_1,...,X_n\}$. Let I be the selected author.
2. We select randomly a text among the 30 available that belong to the I author. Let k be the selected text of author I.
3. We select randomly a number $l \in (L_{min}, L_{max})$.
4. We extract l consecutive lines from text k (starting at the first sentence of the text). Those sentences constitute the generated segment.

Once we have created a dataset, we split it into a training set and a test set, we use the training data to compute μ, σ and optimal γ and r values (by the validation procedure explained in the sequel) and finally run our algorithm on the test data.

Dataset	Authors	No. of docs per set
Set0	Kiosse, Alachiotis	60
Set1	Kiosse, Maronitis	60
Set2	Kiosse, Alachiotis, Maronitis	90
Set3	Kiosse, Alachiotis, Maronitis, Ploritis	120
Set4	Kiosse, Alachiotis, Maronitis, Ploritis, Vokos	150
Dataset	All Authors	300

Table 2. List of the sets complied in the 1st group of experiments using Greek texts and the author's texts used for each of those.

Recall that the segmentation algorithm uses four parameters: μ, σ, r and γ. As already mentioned μ and σ can be interpreted as the average and standard deviation of segment length; it is not immediately obvious how to choose values for r and γ. We use training data and a parameter validation procedure to determine appropriate μ, σ, r and γ values; then we evaluate the algorithm on (previously unseen) test data. More specifically:

1. We choose randomly half of the texts in the dataset to be used as training texts; the rest of the samples are set aside to be used as test texts.
2. We determine appropriate μ and σ values using all the training texts and the standard statistical estimators.
3. We determine appropriate r and γ values by running (on the training texts) the segmentation algorithm with 80 possible combinations of r and γ values; namely we let γ take the 20 values 0.00, 0.01, 0.02, ... , 0.09, 0.1, 0.2, 0.3, ... , 1.0 and let r take the values 0.33, 0.5, 0.66, 1. The optimal (γ , r) combination is the one which yields the minimum P_k value.
4. We apply the algorithm to the test texts using previously estimated μ , σ , r and γ values.

The aforementioned procedure is repeated five times for all sets; the resulting values of Precision, Recall and P_k are averaged. This is performed in order to avoid any problems that can arise from the fact that the various sets of corpus are composed of many segments repeatedly drawn from a small number of different texts. Moreover the fact that texts consisting the training and testing set are randomly selected and the aforementioned procedure is repeated five times, minimizes the probability that a (probably) significant part of the training and testing set is in fact in common. Even this was the case the remaining not common texts would act as "negative" examples i.e. as far as the calculation of the mean and standard deviation is concerned.

In Table 3 we give the values of Precision, Recall and P_k obtained by our algorithm. We also run Choi's and Utiyama's algorithms on the same task; the results are given in Tables 4 and 5.

In Tables 6, 7 and 8 we give the same results averaged over all datasets which have segments of same length. It can be seen that in all cases our algorithm performs significantly better than both Choi's and Utiyama's algorithms. Let us note that the best performance has been achieved for γ in the range [0.08, 0.4] and for r equal to either 0.5 or 0.66.

4.2 Experiment group 2

The second group of experiments also uses Stamatatos's collection; however, the texts are generated using a somewhat different procedure. We constructed a single dataset which contains 200 texts, with every author represented (in other words, the author set is always $\{x_1, x_2, ..., x_{10}\}$). Each text is the concatenation of ten segments. For each segment we do the following:

1. We select randomly an author from $\{x_1, x_2, ..., x_{10}\}$. Let I be the selected author.
2. We select randomly a text among the 30 available that belong to the I author. Let k be the selected text of author I. The selected text is read and scanned in order to determine the number of paragraphs it contains. Let Z be the number of paragraphs that k-th text contains.
3. We select randomly a number $p \in \{1,...,Z\}$ corresponding to the number of paragraphs that the generated segment will contain.
4. We select randomly a number $m \in \{1,..., Z - p\}$ corresponding to the "starting paragraph". Thus the segment contains all the paragraphs of text k starting from paragraph m and ending at the paragraph m + p.

The procedure described above gives texts which are longer than the ones used in Experiment Group 1. Hence the segmentation task in the current group of experiments segmentation of such texts is more difficult than the previous one. Table 9 lists the results we obtained using our algorithm and the ones by Choi and Utiyama. It can be seen again that our algorithm performs better than both Choi's and Utiyama's algorithms.

Dataset	**Precision**	**Recall**	P_k	**Dataset**	**Precision**	**Recall**	P_k
Set01(3-11)	70.65%	71.11%	14.04%	Set31 (3-11)	59.99%	58.67%	17.93%
Set02 (3-5)	86.82%	87.11%	6.20%	Set32 (3-5)	84.44%	83.56%	7.36%
Set03 (6-8)	96.44%	96.44%	0.82%	Set33 (6-8)	86.22%	86.22%	3.28%
Set04(9-11)	93.33%	93.33%	0.84%	Set34 (9-11)	91.11%	91.11%	1.45%
Set11(3-11)	63.86%	67.11%	15.82%	Set41 (3-11)	57.99%	51.11%	17.38%
Set12 (3-5)	82.98%	83.56%	8.47%	Set42 (3-5)	85.00%	84.89%	6.76%
Set13 (6-8)	91.11%	91.11%	2.81%	Set43 (6-8)	88.89%	88.89%	2.65%
Set14(9-11)	94.67%	94.67%	0.98%	Set44 (9-11)	91.11%	91.11%	1.39%
Set21(3-11)	71.14%	60.89%	14.42%	Set51 (3-11)	65.74%	61.78%	14.54%
Set22 (3-5)	90.00%	89.78%	3.45%	Set52 (3-5)	81.56%	81.78%	6.49%
Set23 (6-8)	91.11%	91.11%	2.15%	Set53 (6-8)	89.33%	89.33%	3.57%
Set24(9-11)	92.44	92.44	1.25%	Set54 (9-11)	88.89%	88.89%	1.86%

Table 3. The precision, recall and P_k values obtained by our algorithm for the 1st group of experiments using Greek texts.

Dataset	**Precision**	**Recall**	P_k	**Dataset**	**Precision**	**Recall**	P_k
Set01 (3-11)	65.75%	65.75%	17.06%	Set31 (3-11)	57.75%	57.75%	20.38%
Set02 (3-5)	74.50%	74.50%	16.68%	Set32 (3-5)	70.75%	70.75%	17.40%
Set03 (6-8)	76.50%	76.50%	11.72%	Set33 (6-8)	62.00%	62.00%	17.12%
Set04 (9-11)	64.75%	64.75%	15.08%	Set34 (9-11)	62.00%	62.00%	16.10%
Set11 (3-11)	67.50%	67.50%	16.91%	Set41 (3-11)	57.50%	57.50%	17.38%
Set12 (3-5)	67.75%	67.75%	19.23%	Set42 (3-5)	73.25%	73.25%	15.76%
Set13 (6-8)	72.50%	72.50%	14.74%	Set43 (6-8)	62.50%	62.50%	17.41%
Set14 (9-11)	68.25%	68.25%	14.00%	Set44 (9-11)	63.75%	63.75%	13.70%
Set21 (3-11)	61.00%	61.00%	19.93%	Set51 (3-11)	60.36%	60.50%	17.63%
Set22 (3-5)	73.50%	73.50%	16.15%	Set52 (3-5)	70.50%	70.50%	16.39%
Set23 (6-8)	69.00%	69.00%	15.40%	Set53 (6-8)	67.25%	67.25%	15.85%
Set24 (9-11)	71.75%	71.75%	12.26%	Set54 (9-11)	70.00%	70.00%	12.43%

Table 4. The precision, recall and P_k values obtained by Choi's algorithm for the 1st group of experiments using Greek texts.

Dataset	Precision	Recall	P_k	Dataset	Precision	Recall	P_k
Set01 (3-11)	69.94%	65.55%	15.33%	Set31 (3-11)	61.25%	58.44%	17.64%
Set02 (3-5)	74.16%	59.11%	19.99%	Set32 (3-5)	66.45%	52.88%	20.98%
Set03 (6-8)	80.60%	76.88%	8.94%	Set33 (6-8)	71.88%	70.66%	11.80%
Set04 (9-11)	76.18%	74.45%	8.84%	Set34 (9-11)	67.60%	71.78%	8.75%
Set11 (3-11)	71.41%	68.44%	14.99%	Set41(3-11)	57.77%	56.44%	20.61%
Set12 (3-5)	74.75%	59.11%	18.70%	Set42 (3-5)	71.25%	56.22%	20.07%
Set13 (6-8)	84.77%	83.33%	7.08%	Set43 (6-8)	67.96%	66.44%	12.64%
Set14 (9-11)	81.71%	79.11%	9.10%	Set44 (9-11)	70.23%	72.88%	8.50%
Set21 (3-11)	63.59%	61.11%	18.26%	Set51 (3-11)	60.00%	56.61%	17.41%
Set22 (3-5)	70.57%	53.33%	21.51%	Set52 (3-5)	62.83%	47.55%	23.51%
Set23 (6-8)	77.73%	74.00%	10.75%	Set53 (6-8)	69.56%	66.89%	13.84%
Set24 (9-11)	74.53%	77.33%	7.80%	Set54 (9-11)	68.55%	70.22%	9.99%

Table 5. The precision, recall and P_k values obtained by Utiyama and Isahara's algorithm for the 1st group of experiments using Greek texts.

Dataset	Precision	Recall	P_k
Set*1 (3-11)	64.90%	61.77%	15.69%
Set*2 (3-5)	85.13%	85.11%	6.45%
Set*3 (6-8)	90.51%	90.51%	2.54%
Set*4 (9-11)	91.92%	91.92%	1.29%

Table 6. The precision, recall and P_k values obtained by our algorithm for the 1st group of experiments using Greek texts, averaged over datasets with same-length segments.

Dataset	Precision	Recall	P_k
Set*1 (3-11)	61.64%	61.66%	18.43%
Set*2(3-5)	71.70%	71.70%	16.93%
Set*3 (6-8)	68.29%	68.29%	15.37%
Set*4 (9-11)	66.75%	66.75%	13.93%

Table 7. The precision, recall and P_k values obtained by Choi's algorithm for the 1st group of experiments using Greek texts, averaged over datasets with same-length segments

Dataset	Precision	Recall	P_k
Set*1 (3-11)	64.00%	61.10%	17.37%
Set*2 (3-5)	70.00%	54.70%	20.79%
Set*3 (6-8)	75.42%	73.03%	10.84%
Set*4 (9-11)	73.13%	74.29%	8.83%

Table 8. The precision, recall and P_k values obtained by Utiyama and Isahara's algorithm for the 1st group of experiments using Greek texts, averaged over datasets with same-length segments.

Algorithm	Precision	Recall	P_k
Ours	60.60%	57.00%	11.07%
Choi	44.62%	44.62%	19.44%
Utiyama	56.76%	67.22%	12.28%

Table 9. The precision, recall and P_k values for the 2nd group of experiments using Greek texts.

It is worth mentioning that, the experiments were conducted in a Pentium III 600 MHz with 256 Mbyte RAM memory. The training time of each group was calculated and proved that it is less than two minutes. The average time of calculation for the segmentation of a text by our algorithm was 0.91 seconds.

5. Conclusion

We have presented a text segmentation algorithm following a supervised approach which we applied to the segmentation of Greek texts. On greek text collection our algorithm outperforms Choi's and Utiyama's algorithms. This is largely important particularly in the case of texts exhibiting strong variation as far as the average length is concerned. Let us conclude this paper by discussing the reasons for this performance.

Our algorithm is characterized by (a) the use of dotplot similarity, (b) the form of our similarity function, (c) the use of a length model, (d) the use of dynamic programming, (e) the use of training data. We discuss each of these items in turn.

1. Dotplot similarity. We use a very simple similarity criterion but it is based on the dotplot and hence it captures global similarities, i.e. similarities between every pair of sentences in the document. Dotplots have also been used by Choi (Choi, 2000; Choi et al., 2001), Reynar (Reynar, 1994; Reynar & Ratnaparkhi, 1997) and Xiang and Hongyuan (Xiang & Hongyuan. 2003). On the other hand, Hearst (Hearst, 1994; Hearst & Plaunt, 1993), and Heinonen (Heinonen, 1998) use a cost function which depends only on the similarity of adjacent sentences, hence it is local. Utiyama and Isahara (Utiyama & Isahara, 2001) take an intermediate position: they use a cost function which depends on within-segment statistics, hence it is "somewhat" global, i.e. it considers similarities of all sentences within each segment. Ponte and Croft (Ponte and Croft, 1997) also use an intermediate approach, computing the similarities of all sentences which are at most n sentences apart.
2. Generalized density. We use a very simple similarity function based on a single very simple feature (i.e. we consider sentences similar when they share even a single word). However there is a special characteristic in our function, which we believe to be crucial to the success of our algorithm. Namely, we use the "generalized density" (i.e. $r \neq 2$) and this greatly improves the performance of our algorithm. Other authors have only used dotplot densities with $r = 2$ only (Choi, 2000; Choi et al., 2001; Utiyama & Isahara, 2001; Xiang & Hongyuan, 2003).
3. Length model. A term for the expected length of segments has been used by Ponte and Croft (Ponte and Croft, 1997) and Heinonen (Heinonen, 1998). Utiyama and Isahara (Utiyama & Isahara, 2001) mention the possibility but do not seem to actually use such a model. However, Choi (Choi, 2000; Choi et al., 2001), Reynar (Reynar, 1994; Reynar &

Ratnaparkhi, 1997) and several other authors do not use a length model. We have noticed that the use of the length model greatly enhances the performance of our algorithm.

4. Dynamic programming effects global optimization of the cost function and hence is a very critical factor in the success of our algorithm. As far as we know, the only other authors who have used dynamic programming are Ponte and Croft (Ponte and Croft, 1997), Heinonen (Heinonen, 1998), Xiang (Xiang & Hongyuan, 2003) and, implicitly, Utiyama and Isahara (Utiyama & Isahara, 2001) (their shortest path algorithm is actually a dynamic programming algorithm). On the other hand Choi (Choi, 2000; Choi et al., 2001) and Reynar (Reynar, 1994; Reynar & Ratnaparkhi, 1997) use divisive clustering which, strictly speaking, does not solve an optimization problem; in fact clustering performs a greedy, local optimization. Note also the heuristic approach to segmentation, first used by Hearst (Hearst, 1994; Hearst & Plaunt, 1993) and then by several other authors.
5. Training data. It should not be overlooked that our algorithm depends crucially on the availability of training data, for the estimation of the parameters μ, σ, r and γ. Training data are also used by Choi (Choi, 2000; Choi et al., 2001) for a tuning step of his clustering algorithm; Utiyama and Isahara's algorithm does not depend on training data. However, we should note that in many practical segmentation problems training data will be available (see also (Beeferman et al., 1999)).
6. Finally, for the segmentation of Greek texts we should not overlook the importance of the POS tagger; if the Greek words were not lemmatized, the vocabulary of the text collection would increase by an order of magnitude, making the segmentation problem much harder.

In short, we believe that our algorithm outperforms Choi's and Utiyama's algorithms because it performs global optimization of a global cost function. This should be compared to the local optimization of global information (used by Choi) and the global optimization of local information (used by Utiyama and Isahara).

In future work, we plan to apply our dynamic programming method to other similarity metrics such as the one proposed by Hearst (WindowDiff) in order to assess the difference in segmentation accuracy.

An interesting point would be to test our algorithm in text of continuous stream i.e. longer texts than the one used for the second experiment for the greek texts. Another interesting point to examine is to enhance the vector space model used in order to calculate the similarity between sentences with the ranking (3x3 grid which is roughly equal to the one common word measure) measure in order to avoid any stability issues that may rise by the similarity metric used by our algorithm.

In order to combine our algorithm with psychological issues such as the words used by different authors, we plan to examine some of the at least well known 1000 textual attributes relevant to authorship. The selection of those variables is based on their ability to reveal subconscious mechanisms of language variation which are unique to each author and have an impact on the discrimination of the author among every possible author, thus in our case, topic i.e. segment change. As it was proposed by Bestgen (Bestgen, 2006) our algorithm can benefit from the addition of semantic knowledge for capturing semantic relations between words appearing in sentences, which will be a future step.

6. References

Auran, C. ; Colas, A. ; Portes, C. & Vion, M. (2005). Perception of breaks and discourse boundaries in spontaneous speech: developing an on-line technique. *IDP05 - Discours et Prosodie comme Interface Complexe*.

Beeferman, D.; Berger, A. & Lafferty, J. (1999). Statistical models for text segmentation. *Machine Learning*, vol. 34, pp. 177-210.

Beeferman, D.; Berger, A. & Lafferty, J. (1997). Text segmentation using exponential models. *In Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pp. 35-46.

Bertsekas, D. (1987). Dynamic Programming: Deterministic and Stochastic Models. *Prentice Hall*.

Bestgen, Y. (2006). Improving Text Segmentation Using Latent Semantic Analysis: A Reanalysis of Choi, Wiemer-Hastings Deterministic and Moore (2001). *Computational Linguistics*, vol. 1, pp. 5-12.

Choi, F.Y.Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 26-33.

Choi, F.Y.Y.; Wiemer-Hastings, P. & Moore, J. (2001). Latent semantic analysis for text segmentation. In *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing*, pp. 109-117.

Dias, G. & Alves, E. (2005). Unsupervised Topic Segmentation Based on Word Cooccurrence and Multi-Word Units for Text Summarization. E*LECTRA Workshop Methodologies and Evaluation of Lexical Cohesion Techniques in Real-world Applications Beyond Bag of Words (in association with SIGIR-2005)*, pp. 41-48.

Dowman, M.; Tablan, V.; Cunningham, H.; Ursu, C. & Popov, B. (2005). Semantically Enhanced Television News through Web and Video Integration. In *Proceedings of the ESWC05 Workshop on Multimedia and the Semantic Web*.

Halliday, M. & Hasan, R. (1976). Cohesion in English. *Longman*.

Hearst, M. A. (1994). Multi-paragraph segmentation of expository texts. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistic*, pp. 9-16.

Hearst, M. A. & Plaunt, C. (1993). Subtopic structuring for full-length document access. In *Proceedings of the 16th Annual International Conference on Research and Development in information Retrieval of the Association of Computer Machinery - Special Interest Group on Information Retrieval (ACM-SIGIR)*, pp. 59-68.

Heinonen, O. (1998). Optimal Multi-Paragraph Text Segmentation by Dynamic Programming. In *Proceedings of 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pp. 1484-1486.

Hirschberg, J. & Litman, D. (1993). Empirical studies on the disambiguation and cue phrases. *Computational Linguistics*, vol.19, pp. 501-530.

Hsueh, P-Y.; Moore, J.D. & Renals, S. (2006). Automatic Segmentation of Multiparty Dialogue. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL) 2006*, pp. 273-280.

Kehagias, Ath.; Nicolaou A. ; Fragkou P. & Petridis V. (2004)(a). Text Segmentation by Product Partition Models and Dynamic Programming. *Mathematical and Computer Modeling*, vol. 39, pp. 209-217.

Kehagias, Ath.; Fragkou P. & Petridis V. (2004)(b). A Dynamic Programming Algorithm for Linear Text Segmentation. *Journal of Int. Information Systems*, vol. 23, pp. 179-197.

Kiat T.Y. (2005). Linear and Hierarchical Text Segmentation Using Product Partition Models. Master Thesis, Department of Computer Science, School of Computing, National University of Singapore 2004/2005.

Koshinaka, T.; Iso, K.-I. & Okumura, A. (2005). An HMM-based text segmentation method using variational Bayes approach and its application to LVCSR for broadcast news. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.1, pp. 485- 488.

Kozima, H. (1993). Text Segmentation based on similarity between words. *In Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 286-288.

Kozima, H & Furugori, T. (1993). Similarity between words computed by spreading activation on an English dictionary. *In Proceedings of 6th Conference of the European Chapter of the Association or Computational Linguistics*, pp. 232-239.

McDonald, R.; Crammer, K. & Pereira, F. (2005). Flexible Text Segmentation with Structured Multilabel Classification. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), Association for Computational Linguistics*, pp. 987–994.

Morris, J. & Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, vol. 17, pp. 21-42.

Olney, A. & Cai, Z. (2005). An Orthonormal Basis for Topic Segmentation in Tutorial Dialogue. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005, AAAI Press, pp. 971-978.

Orphanos, G. & Christodoulakis, D. (1999). Part-of-speech disambiguation and unknown word guessing with decision trees. In *Proceedings of EACL'99*.

Orphanos, G. & Tsalidis, C. (1999). Combining handcrafted and corpus-acquired lexical knowledge into a morphosyntactic tagger. In *Proceedings of the 2nd Research Colloquium for Computational Linguistics in United Kingdom (CLUK).*

Passoneau, R. & Litman, D.J. (1993). Intention - based segmentation: Human reliability and correlation ith linguistic cues. *In Proceedings of the 31st Meeting of the Association for Computational Linguistics*, pp. 148-155.

Pevzner, L. & Hearst, M. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, vol.28(1), pp. 19–36.

Ponte, J. M. & Croft, W. B. (1997). Text segmentation by topic. In *Proceedings of the 1st European Conference on Research and Advanced Technology for Digital Libraries*, pp. 120 - 129.

Raskin, V., & Weiser, J. (1987). Language and Writiing: Applications of linguistics to rhetoric and composition. *Norwood, New Jersey: ABLEX: Publishing Corporation*.

Reynar, J.C. (1994). An automatic method of finding topic boundaries. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 331-333.

Reynar, J.C. & Ratnaparkhi, A. (1997). A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pp. 16-19.

Roget, P.M. (1977). *Roget's International Thesaurus*. Harper and Row, 4th edition.

Sitbon, L. & Bellot, P. (2005). Segmentation thématique par chaînes lexicales pondérées. Actes de TALN 2005, Dourdan, France.

Stamatatos, E.; Fakotakis, N. & Kokkinakis, G. (2001). Computer-based authorship attribution without lexical measures. *Computer and the Humanities*, Kluwer Academic Publishers, vol. 35, pp. 193 - 214.

Utiyama, M., & Isahara, H. (2001). A statistical model for domain - independent text segmentation. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 491-498.

Ursu, C.; Tablan, V. & Cunningham, H. (2005). Semantic Analysis for tomorrow's audio-visual digital archives. In *Proceedings of 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies (EWIMT-2005)*.

Xiang J. & Hongyuan Z. (2003). Domain-independent Text Segmentation Using Anisotropic Diffusion and Dynamic Programming. In *Proceedings of the 26th ACM SIGIR Conference. on Research and Development in Information Retrieval*.

Yaari, Y. (1997). Segmentation of expository texts by hierarchical agglomerative clustering. In *Proceedings of the Conference on Recent Advances in Natural Language Processing*, pp. 59-65.

Yaari, Y. (1999). Intelligent exploration of expository texts. Ph.D. thesis. Dept. of Computer Science, Bar-Ilan University.

Ye, N.; Zhu, J.; Luo, H.; Wang, H. & Zhang, B. (2005). Improvement of the dotplotting method for linear text segmentation. *Natural Language Processing and Knowledge Engineering*, pp. 636- 641.

Appendix A: The Morphosyntactic Tagger

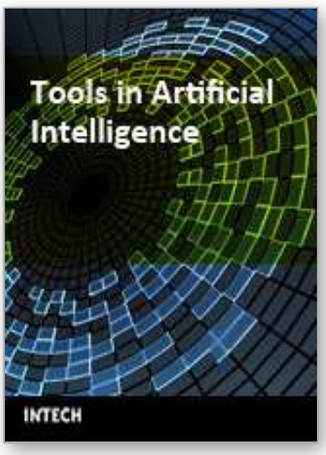
The Greek texts were preprocessed using the morphosyntactic tagger (better known as Part-Of-Speech tagger) developed by Ophanos et al. (Orphanos & Christodoulakis, 1999; Orphanos & Tsalidis, 1999). This is a Part-Of-Speech (POS) tagger for modern Greek (a high inflectional language) and is based on a Lexicon capable of assigning full morphosyntactic attributes (i.e. Part-Of-Speech, Number, Gender, Tense, Voice, Mood and Lemma) to 876.000 Greek word forms. Orphanos et al. created a tagged corpus capable of exhibiting the capability of the POS tagger to identify and resolve all POS ambiguity schemes present in Modern Greek (e.g. Pronoun-Clitic-Article, Pronoun-Clitic, Adjective-Adverb, Verb-Noun, etc) as well as the characteristics of unknown words by using the Lexicon. They used this corpus in order to induce decision trees, which along with the Lexicon are integrated into a robust POS tagger for Modern Greek texts. The tagger has three parts: the Tokenizer, the Lexicon and finally the Disambiguator and Guesser. The Tokenizer takes as input raw text and converts it into a stream of tokens. The Tokenizer resolves non-word tokens (e.g. punctuation marks, numbers, dates etc.) and provides them a tag corresponding to their category. As for the word tokens, they are looked up in the Lexicon and those found receive one or more tags. The Disambiguator/Guesser takes as input words that received more than one tags and words that were not found in the Lexicon and decides their contextually appropriate tag. The Disambiguator/Guesser is a 'forest' of decision trees, one tree for each ambiguity scheme present in Modern Greek and one tree for unknown guessing. The ambiguity scheme of words that received by the Lexicon more than one tag is identified and the corresponding decision tree is selected. This tree is traversed according to the values of the morphosyntactic features extracted from contextual tags. The result of this traversal is

the contextually appropriate POS tag along with its corresponding lemma. In order to resolve the ambiguity, tag(s) with different POS than the one returned by the decision tree, is (are) eliminated. In order to determine the POS of an unknown word, the decision tree for unknown words is traversed and examines contextual features along with the word ending and capitalization. As a result the open class POS and the corresponding lemma of the unknown word are returned.

Appendix B

<CC>
Γ. ΔΕΡΤΙΛΗΣ ΤΟ ΒΗΜΑ, 23-03-1997 Κωδικός άρθρου: Β12421Β062 </CC>
<TITLE>
Σαφήνεια και αμφιβολία
</TITLE>
<TEXT>
Προϋπόθεση του καλού ύφους, η σαφήνεια είναι αναγκαία τόσο στη λογοτεχνία όσο και στην επιστημονική γραφή. Αλλά πρόκειται για δύο διαφορετικές σαφήνειες. Η μία είναι ποιητική, η άλλη εξηγηματική.
Με τη σαφήνεια του ύφους του, ο λογοτέχνης «ποιεί» την πολυσημία. Ετσι ανοίγει μπροστά στον αναγνώστη ένα ριπίδιο αναγνώσεων: τον ευκολύνει να διαβάσει και να ερμηνεύσει το πολύσημο κείμενο με πολλαπλούς τρόπους.
Αλλά ο συγγραφέας ενός επιστημονικού έργου (αυτός που κυρίως θα μας απασχολήσει σήμερα) εξαφανίζει με τη σαφήνεια του ύφους του όλες τις αμφισημίες και πολυσημίες του κειμένου. Αποκλείει έτσι τις αμφιβολίες του αναγνώστη για τα όσα ο συγγραφέας ισχυρίζεται και διευκολύνει τον ανα-γνωστικό, επιστημονικό έλεγχο. Η πολυσημία που προσπαθεί να εκφράσει ο λογοτέχνης μοιάζει, εξάλλου, αλλά δεν ταυτίζεται με την αμφιβολία που κάποτε εκφράζει στο κείμενό του ένας επιστήμονας. Την εκφράζει επειδή συναισθάνεται τα όρια του εαυτού του, του συγκεκριμένου έργου του, των προσωπικών του θεωριών, ακόμη και της επιστήμης του. Αλλά παραμένει η ανάγκη να είναι σαφείς οι θεωρίες του, σαφές και το κείμενό του. Ετσι, ο συγγραφέας από τη μια καταγράφει την αμφιβολία, από την άλλη όμως υποστηρίζει με σαφήνεια τη συλλογιστική του, τις απόψεις και τις ερμηνείες του: επειδή ο επιστημονικός λόγος, εξ ορισμού, δεν επιδέχεται αντιφάσεις.
Οπως είναι φυσικό, ο κανόνας της σαφήνειας δεν έχει ενιαία εφαρμογή. Υπάρχουν οι διαφοροποιήσεις που εξαρτώνται από την προσωπικότητα και τις ικανότητες του κάθε συγγραφέα. Ενας επιστήμονας με καλό συγγραφικό ταλέντο μπορεί ίσως να βρει ελευθεριότερους τρόπους παρουσίασης των ιδεών του, να επεκταθεί σε υπαινιγμούς, σε αμφισημίες και σε αποσιωπήσεις που έχουν τη δική τους λειτουργία και αισθητική. Αλλά αυτό δεν αναιρεί την επιστημονική του υποχρέωση να δείξει με σαφήνεια, σε άλλα σημεία του κειμένου, τις απόψεις και τις ερμηνείες του.
Υπάρχουν έπειτα διαφοροποιήσεις ανάλογες με τα γνωστικά αντικείμενα και τα είδη του γραπτού επιστημονικού λόγου. Η Ιστορία, π.χ., αφήνει περισσότερες υφολογικές δυνατότητες στον συγγραφέα από οποιαδήποτε άλλη επιστήμη. Του επιτρέπει, κάποτε του επιβάλλει κιόλας, να αναδείξει τις εσωτερικές αντιφάσεις του ανθρώπου και των ανθρωπίνων κοινωνιών· τον ρόλο των ανθρωπίνων παθών· τη σημασία των συμπτώσεων και της τύχης· το βάρος των μαζικών κοινωνικών δυνάμεων· τους

αναπόδραστους φραγμούς της φύσης.
Ωστόσο, ο ιστορικός δεν δείχνει τις αντιφάσεις ουσίας με αντιφάσεις ύφους, αλλά με σαφήνεια. Τα πάθη δεν τα δείχνει με ψευδορομαντική ασάφεια, αλλά με τη σαφήνεια εκείνη που θα αναδείξει την αιχμηρότητά τους. Τονίζει τις συμπτώσεις και την τυχαιότητα με ύφος σαφές και όχι τυχάρπαστο. Τη «μοίρα» δεν την αποδίδει σε μεταφυσικές δυνάμεις - εφόσον κάνει επιστήμη. Μπορεί να την ταυτίζει με δυνάμεις που θεωρούσαν ανεξήγητες και μεταφυσικές οι άνθρωποι που μελετά· αλλά ο ίδιος δίνει όνομα στις δυνάμεις αυτές· και τις εντάσσει, με σαφήνεια, σε έναν αιτιακό συλλογισμό, σε ένα ερμηνευτικό σχήμα.
Ενα ευτυχές ιστοριογραφικό έργο απαιτεί έναν καλό συγκερασμό της επιστήμης με την τέχνη του ύφους. Από εκεί και πέρα, υπάρχει μόνο η υπέρβαση και της επιστήμης και του ύφους. Στον υπερβατικό αυτό χώρο, εκεί όπου ο συγκερασμός γίνεται ταύτιση γνώσης και τέχνης, οδηγεί ένας δρόμος σχεδόν άβατος. Τόπος που ονειρεύονται πολλοί, επιστήμονες και τεχνίτες, τόπος άφθαστος για μας τους πολλούς - όχι, όμως, ουτοπία. Μας τον έχουν δείξει οι ελάχιστοι που έφτασαν εκεί, οι δάσκαλοί μας, ο καθένας με τη μεγάλη και τη μικρή του ιστορία, όντα διόλου μεταφυσικά, πολύ ανθρώπινα. Ενας απλός άνθρωπος δεν ήταν άραγε ο δάσκαλος που, πριν από δυόμισι αιώνες, σκάρωνε κάθε μέρα τη φυγή του προς τα εκεί, με ένα απλό, αλλά καλώς συγκερασμένο κλειδοκύμβαλο;
Ο κ. Γ. Β. Δερτιλής είναι καθηγητής της Ιστορίας στο Πανεπιστήμιο Αθηνών.
</TEXT>



Tools in Artificial Intelligence

Edited by Paula Fritzsche

ISBN 978-953-7619-03-9
Hard cover, 488 pages
Publisher InTech
Published online 01, August, 2008
Published in print edition August, 2008

This book offers in 27 chapters a collection of all the technical aspects of specifying, developing, and evaluating the theoretical underpinnings and applied mechanisms of AI tools. Topics covered include neural networks, fuzzy controls, decision trees, rule-based systems, data mining, genetic algorithm and agent systems, among many others. The goal of this book is to show some potential applications and give a partial picture of the current state-of-the-art of AI. Also, it is useful to inspire some future research ideas by identifying potential research directions. It is dedicated to students, researchers and practitioners in this area or in related fields.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen