Automatic Paragraph Identification: A Study across Languages and Domains

Caroline Sporleder
School of Informatics
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW, UK
csporled@inf.ed.ac.uk

Mirella Lapata
Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield S1 4DP, UK
mlap@dcs.shef.ac.uk

Abstract

In this paper we investigate whether paragraphs can be identified automatically in different languages and domains. We propose a machine learning approach which exploits textual and discourse cues and we assess how well humans perform on this task. Our best models achieve an accuracy that is significantly higher than the best baseline and, for most data sets, comes to within 6% of human performance.

1 Introduction

Written texts are usually broken up into sentences and paragraphs. Sentence splitting is a necessary pre-processing step for a number of Natural Language Processing (NLP) tasks including part-of-speech tagging and parsing. Since sentence-final punctuation can be ambiguous (e.g., a period can also be used in an abbreviation as well as to mark the end of a sentence), the task is not trivial and has consequently attracted a lot of attention (e.g., Reynar and Ratnaparkhi (1997)). In contrast, there has been virtually no previous research on inferring paragraph boundaries automatically. One reason for this is that paragraph boundaries are usually marked unambiguously by a new line and extra white space.

However, a number of applications could benefit from a paragraph detection mechanism. Text-to-text generation applications such as single- and multidocument summarisation as well as text simplification usually take naturally occurring texts as input and transform them into new texts satisfying specific constraints (e.g., length, style, language). The output texts do not always preserve the structure and editing conventions of the original text. In summarisation, for example, sentences are typically extracted verbatim and concatenated to form a summary. Insertion of paragraph breaks could improve the readability of the summaries by indicating topic shifts and providing visual targets to the reader (Stark, 1988).

Machine translation is another application for which automatic paragraph detection is relevant. Current systems deal with paragraph boundary insertion in the target language simply by preserving the boundaries from the source language. However, there is evidence for cross-linguistic variation in paragraph formation and placement, particularly for languages that are not closely related such as English and Chinese (Zhu, 1999). So, a paragraph insertion mechanism that is specific to the target language, instead of one that relies solely on the source language, may yield more readable texts.

Paragraph boundary detection is also relevant for speech-to-text applications. The output of automatic speech recognition systems is usually raw text without any punctuation or paragraph breaks. This naturally makes the text very hard to read, which can cause processing difficulties, especially if speech recognition is used to provide deaf students with real-time transcripts of lectures. Furthermore, sometimes the output of a speech recogniser needs to be processed automatically by applications such as information extraction or summarisation. Most of these applications (e.g., Christensen et al., (2004)) port techniques developed for written texts to spoken texts and therefore require input that is punctuated and broken into paragraphs. While there has been some research on finding sentence boundaries in spoken text (Stevenson and Gaizauskas, 2000), there has been little research on determining paragraph boundaries.[1]

If paragraph boundaries were mainly an aesthetic device for visually breaking up long texts into smaller chunks, as has previously been suggested (see Longacre (1979)), paragraph boundaries could be easily inserted by splitting a text into several equal-size segments. Psycho-linguistic research, however, indicates that paragraph boundaries are not purely aesthetic. For example, Stark (1988)

[1] There has been research on using phonetic cues to segment speech into "acoustic paragraphs" (Hauptmann and Smith, 1995). However, these do not necessarily correspond to written paragraphs. But even if they did, textual cues could complement phonetic information to identify paragraphs.

asked her subjects to reinstate paragraph boundaries into fiction texts from which all boundaries had been removed and found that humans are able to do so with an accuracy that is higher than would be expected by chance. Crucially, she also found that (a) individual subjects did not make all their paragraphs the same length and (b) paragraphs in the original text whose length deviated significantly from the average paragraph length were still identified correctly by a large proportion of subjects. These results show that people are often able to identify paragraphs correctly even if they are exceptionally short or long without defaulting to a simple template of average paragraph length.

Human agreement on the task suggests that the text itself provides cues for paragraph insertion, even though there is some disagreement over which specific cues are used by humans (see Stark (1988)). Possible cues include repeated content words, pronoun coreference, paragraph length, and local semantic connectedness.

In this paper, we investigate whether it is possible to exploit some of these textual cues together with syntactic and discourse related information to determine paragraph boundaries automatically. We treat paragraph boundary identification as a classification task and examine whether the difficulty of the task and the utility of individual textual cues varies across languages and across domains. We also assess human performance on the same task and whether it differs across domains.

2 Related Work

Previous work has focused extensively on the task of automatic text segmentation whose primary goal is to divide individual texts into sub-topics. Despite their differences, most methods are unsupervised and typically rely on the distribution of words in a given text to provide cues for topic segmentation.[2] Hearst's (1997) TextTiling algorithm, for example, determines sub-topic boundaries on the basis of term overlap in adjacent text blocks. In more recent work, Utiyama and Isahara (2001) combine a statistical segmentation model with a graph search algorithm to find the segmentation with the maximum probability. Beeferman et al. (1999) use supervised learning methods to infer boundaries between texts. They employ language models to detect topic shifts and combine them with cue word features.

Our work differs from these previous approaches in that paragraphs do not always correspond to subtopics. While topic shifts often correspond to paragraph breaks, not all paragraph breaks indicate a topic change. Breaks between paragraphs are often inserted for other (not very well understood) reasons (see Stark (1988)). Therefore, the segment granularity is more fine-grained for paragraphs than for topics. An important advantage for methods developed for paragraph detection (as opposed to those developed for text-segmentation) is that training data is readily available, since paragraph boundaries are usually unambiguously marked in texts. Hence, supervised methods are "cheap" for this task.

3 Our Approach

3.1 Corpora

Our study focused on three languages: English, German, and Greek. These languages differ in terms of word order (fixed in English, semi-free in German, fairly flexible in Greek). Greek and German also have richer morphology than English. Additionally, Greek has a non-Latin writing system.

For each language we created corpora representative of three domains: fiction, news, and parliamentary proceedings. Previous work on the role of paragraph markings (Stark, 1988) has focused exclusively on fiction texts, and has shown that humans can identify paragraph boundaries in this domain reliably. It therefore seemed natural to test our automatic method on a domain for which the task has been shown to be feasible. We selected news texts since most summarisation methods today focus on this domain and we can therefore explore the relevance of our approach for this application. Finally, parliamentary proceedings are transcripts of speech, and we can examine whether a method that relies solely on textual cues is also useful for spoken texts.

For English, we used the whole Hansard section of the BNC, as our corpus of parliamentary proceedings. We then created a fiction corpus of similar size by randomly selecting prose files from the fiction part of the BNC. In the same way a news corpus was created from the Penn Treebank.

For German, we used the prose part of Project Gutenberg's e-book collection[3] as our fiction corpus and the complete Frankfurter Rundschau part of the ECI corpus[4] as our news corpus. The corpus of parliamentary proceedings was obtained by randomly

[2] Due to lack of space we do not describe previous work in text segmentation here in detail; we refer the reader to Utiyama and Isahara (2001) and Pevzener and Hearst (2002) for a comprehensive overview.

[3] http://www.gutenberg.net/ For copyright reasons, this web site mainly contains books published before 1923.

[4] http://www.elsnet.org/eci.html

	fiction	news	parliament
English	1,140,000	1,156,000	1,156,000
German	2,500,000	4,100,000	3,400,000
Greek	563,000	1,500,000	1,500,000

Table 1: Number of words per corpus

selecting a subset of the German section from the Europarl corpus (Koehn, 2002).

For Greek, a fiction corpus was compiled from the ECI corpus by selecting all prose files that contained paragraph markings. Our news corpus was downloaded from the WWW site of the Modern Greek newspaper Eleftherotypia and consists of financial news from the period of 2001–2002. A corpus of parliamentary proceedings was again created by randomly selecting a subset of the Greek section of the Europarl corpus (Koehn, 2002).

Parts of the data were further pre-processed to insert sentence boundaries. We trained a publicly available sentence splitter (Reynar and Ratnaparkhi, 1997) on a small manually annotated sample (1,000 sentences per domain per language) and applied it to our corpora. Table 1 shows the corpus sizes. All corpora were split into training (72%), development (24%) and test set (4%).

3.2 Machine Learning

We used BoosTexter (Schapire and Singer, 2000) as our machine learning system. BoosTexter was originally developed for text categorisation and combines a boosting algorithm with simple decision rules. For all domains and languages our training examples were sentences. Class labels encoded for each sentence whether it was starting a paragraph or not.

The features we used fall broadly into three different areas: non-syntactic features, language modelling features and syntactic features. The latter were only applied to English as we did not have suitable parsers for German and Greek.

The values of our features are numeric, boolean or "text". BoosTexter applies unigram models when forming classification hypotheses for features with "text" values. These can be simply words or annotations such as part-of-speech tags.

We deliberately did not include anaphora-based features. While anaphors can help determine paragraph boundaries (paragraph initial sentences tend to contain few or no anaphors), anaphora structure is dependent on paragraph structure rather than the other way round. Hence, in applications which manipulate texts and thereby potentially "mess-up" the anaphora structure (e.g., multi-document summarisation), anaphors are not a reliable cue for paragraph identification.[5]

3.2.1 Non-syntactic Features

Distance (D_s, D_w): These features encode the distance of the current sentence from the previous paragraph break. We measured distance in terms of the number of intervening sentences (D_s) as well as in terms of the number of intervening words (D_w). If paragraph breaks were driven purely by aesthetics one would expect this feature to be among the most successful ones.[6]

Sentence Length (*Length*): This feature encodes the number of words in the current sentence. Average sentence length is known to vary with text position (Genzel and Charniak, 2003) and it is possible that it also varies with paragraph position.

Relative Position (*Pos*): The relative position of a sentence in the text is calculated by dividing the current sentence number by the number of sentences in the text. The motivation for this feature is that paragraph length may vary with text position. For example, it is possible that paragraphs at the beginning and end of a text are shorter than paragraphs in the middle and hence a paragraph break is more likely at the two former text positions.

Quotes ($Quote_p$, $Quote_c$, $Quote_i$): These features encode whether the previous or current sentence contain a quotation ($Quote_p$ and $Quote_c$, respectively) and whether the current sentence continues a quotation that started in a preceding sentence ($Quote_i$). The presence of quotations can provide cues for speaker turns, which are often signalled by paragraph breaks.

Final Punctuation (*FinPun*): This feature keeps track of the final punctuation mark of the previous sentence. Some punctuation marks may provide hints as to whether a break should be introduced. For example, in the news domain, where there is hardly any dialogue, if the previous sentence ended in a question mark, it is likely that the current sentence supplies an answer to this question, thus making a paragraph break improbable.

Words (W_1, W_2, W_3, W_{all}): These text-valued features encode the words in the sentence. W_{all} takes the complete sentence as its value. W_1, W_2 and W_3 encode the first word, the first two words and the first three words, respectively.

[5] This is also true for some of the other features we use (e.g., sentence length) but not quite to the same extent.

[6] One could also use the history of class labels assigned to previous sentences as a feature (as in part-of-speech tagging); however, we leave this to future research.

3.2.2 Language Modelling Features

Our motivation for including language modelling features stems from Genzel and Charniak's (2003) work where they show that the word entropy rate is lower for paragraph initial sentences than for non-paragraph initial ones. We therefore decided to examine whether word entropy rate is a useful feature for the paragraph prediction task. Using the training set for each language and domain, we created language models with the CMU language modelling toolkit (Clarkson and Rosenfeld, 1997). We experimented with language models of variable length (i.e., 1–5) and estimated two features: the probability of a given sentence according to the language model (LM_p) and the per-word entropy rate (LM_{pwe}). The latter was estimated by dividing the sentence probability as assigned by the language model by the number of sentence words (see Genzel and Charniak (2003)).

We additionally experimented with character level *n*-gram models. Such models are defined over a relatively small vocabulary and can be easily constructed for any language without pre-processing. Character level *n*-gram models have been applied to the problem of authorship attribution and obtained state-of-the art results (Peng et al., 2003). If some characters are more often attested in paragraph starting sentences (e.g., "A" or "T"), then we expect these sentences to have a higher probability compared to non-paragraph starting ones. Again, we used the CMU toolkit for building the character level *n*-gram models. We experimented with models whose length varied from 2 to 8 and estimated the probability assigned to a sentence according to the character level model (CM_p).

3.2.3 Syntactic Features

For the English data we also used several features encoding syntactic complexity. Genzel and Charniak (2003) suggested that the syntactic complexity of sentences varies with their position in a paragraph. Roughly speaking, paragraph initial sentences are less complex. Hence, complexity measures may be a good indicator of paragraph boundaries. To estimate complexity, we parsed the texts with Charniak's (2001) parser and implemented the following features:

Parsed: This feature states whether the current sentence could be parsed. While this is not a real measure of syntactic complexity it is probably correlated with it.

Number of phrases (num_s, num_{vp}, num_{np}, num_{pp}): These features measure syntactic complexity in terms of the number of S, VP, NP, and PP constituents in the parse tree.

Signature ($Sign$, $Sign_p$): These text-valued features encode the sequence of part-of-speech tags in the current sentence. *Sign* only encodes word tags, while $Sign_p$ also includes punctuation tags.

Children of Top-Level Nodes ($Childr_{s1}$, $Childr_s$): These text-valued features encode the top-level complexity of a parse tree: $Childr_{s1}$ takes as its value the sequence of syntactic labels of the children of the S1-node (i.e., the root of the parse tree), while $Childr_s$ encodes the syntactic labels of the children of the highest S-node(s). For example, $Childr_{s1}$ may encode that the sentence consists of one clause and $Childr_s$ may encode that this clause consists of an NP, a VP, and a PP.

Branching Factor ($Branch_s$, $Branch_{vp}$, $Branch_{np}$, $Branch_{pp}$): These features express the average number of children of a given non-terminal constituent (cf. Genzel and Charniak (2003)). We compute the branching factor for S, VP, NP, and PP constituents.

Tree Depth: We define tree depth as the average length of a path (from root node to leaf node).

Cue Words (Cue_s, Cue_m, Cue_e): These features are not strictly syntactic but rather discourse-based. They encode discourse cues (such as *because*) at the start (Cue_s), in the middle (Cue_m) and at the end (Cue_e) of the sentence, where "start" is the first word, "end" the last one, and everything else is "middle". We keep track of all cue word occurrences, without attempting to distinguish between their syntactic and discourse usages.

For English, there are extensive lists of discourse cues (we used Knott (1996)), but such lists are not widely available for German and Greek. Hence, we only used this feature on the English data.

4 Experiments

BoosTexter is parametrised with respect to the number of training iterations. In all our experiments, this parameter was optimised on the development set; BoosTexter was initially trained for 500 iterations, and then re-trained with the number of iterations that led to the lowest error rate on the development set. Throughout this paper all results are reported on the unseen test set and were obtained using models optimised on the development set. We report the models' accuracy at predicting the right label (i.e., paragraph starting or not) for each sentence.

	English			German			Greek		
feature	fiction	news	parl.	fiction	news	parl.	fiction	news	parl.
B_d	60.16	51.73	59.50	65.44	59.03	58.26	59.00	52.85	66.48
B_m	71.04	51.44	69.38	75.75	68.24	66.17	67.57	53.99	76.25
$Dist_s$	71.07	57.74	54.02	75.80	68.25	66.23	67.69	57.94	76.30
$Dist_w$	71.02	**63.08**	65.64	75.80	67.70	67.20	**68.31**	59.76	76.30
$Length$	72.08	56.11	68.45	75.75	72.55	67.10	67.52	60.84	76.55
$Position$	71.04	49.18	38.71	75.68	68.05	66.35	67.57	56.52	76.35
$Quote_p$	**80.84**	56.25	30.62	72.97	68.24	66.23	**72.80**	58.00	76.30
$Quote_c$	**80.64**	54.95	31.00	72.35	68.24	66.17	71.03	53.99	76.25
$Quote_i$	71.04	51.44	30.62	75.75	68.24	66.17	67.57	53.99	76.25
$FinPun$	72.08	54.18	71.75	73.15	**76.36**	69.53	**73.33**	59.86	76.55
W_1	72.96	57.74	**82.05**	75.43	73.87	75.25	67.05	67.41	76.81
W_2	73.47	58.51	**80.62**	75.80	**74.77**	**76.74**	66.37	**68.22**	**78.48**
W_3	73.68	**59.90**	**80.73**	75.60	**74.50**	**76.79**	67.63	**67.88**	**78.43**
W_{all}	**73.99**	**61.78**	75.40	75.60	73.03	**76.20**	67.78	**67.88**	**77.26**
$BestLM_p$	72.83	55.96	69.66	**75.93**	71.39	67.40	67.57	61.64	76.50
$BestLM_{pwe}$	72.16	52.21	69.88	**75.90**	69.24	66.98	67.83	56.29	76.40
$BestCM_p$	72.70	57.36	69.49	**75.88**	73.37	67.53	67.68	61.68	76.51
all_{ns_lcm}	**82.45**$*$	**70.77**$*$	**82.71**$*$	**76.55**$\not*$	**79.28**$*$	**79.17**$*$	**78.03**$*$	**76.31**$*$	**79.35**$*$

Table 2: Accuracy of non-syntactic and language modelling features on test set

4.1 The Influence of Non-syntactic Features

In the first set of experiments, we ran BoosTexter on all 9 corpora using non-syntactic and language modelling features. To evaluate the contribution of individual features to the classification task, we built one-feature classifiers in addition to a classifier that combined all features. Table 2 shows the test set classification accuracy of the individual features and their combination (all_{ns_lcms}). The length of the language and character models was optimised on the development set. The test set accuracy of the optimised models is shown as $BestLM_p$ and $BestLM_{pwe}$ (language models) and $BestCM_p$ (character models).[7] The results for the three best performing one-feature classifiers and the combined classifier are shown in boldface.

BoosTexter's classification accuracy was further compared against two baselines. A distance-based baseline (B_d) was obtained by hypothesising a paragraph break after every d sentences. We estimated d in the training data by counting the average number of sentences between two paragraphs. Our second baseline, B_m, defaults to the majority class, i.e., assumes that the text does not have paragraph breaks.

For all languages and domains, the combined models perform better than the best baseline. In order to determine whether this difference is significant, we applied χ^2 tests. The diacritic $*$ ($\not*$) in Table 2 indicates whether a given model is (not) significantly different from the best baseline. Significant results are achieved across the board with the exception of German fiction. We believe the reason for this lies in the corpus itself, as it is very heterogeneous, containing texts whose publication date ranges from 1766 to 1999 and which exhibit a wide variation in style and orthography. This makes it difficult for any given model to reliably identify paragraph boundaries in all texts.

In general, the best performing features vary across domains but not languages. Word features (W_1–W_3, W_{all}) yield the best classification accuracies for news and parliamentary domains, whereas for fiction, quotes and punctuation seem more useful. The only exception is the German fiction corpus, which consists mainly of 19th century texts. These contain less direct speech than the two fiction corpora for English and Greek (which contain contemporary texts). Furthermore, while examples of direct speech in the English corpus often involve short dialogues, where a paragraph boundary is introduced after each speaker turn, the German corpus contains virtually no dialogues and examples of direct speech are usually embedded in a longer narrative and not surrounded by paragraph breaks.

Note that the distance in words from the previous paragraph boundary ($Dist_w$) is a good indicator for a paragraph break in the English news domain. However, this feature is less useful for the other two

[7] Which language and character models perform best varies slightly across corpora but no clear trends emerge.

languages. An explanation might be that the English news corpus is very homogeneous (i.e., it contains articles that not only have similar content but are also structurally alike). The Greek news corpus is relatively homogeneous; it mainly contains financial news articles but also some interviews, so there is greater variation in paragraph length, which means that the distance feature is overtaken by the word-based features. Finally, the German news corpus is highly heterogeneous, containing not only news stories but also weather forecasts, sports results and cinema listings. This leads to a large variation in paragraph length, which in turn means that the distance feature performs worse than the best baseline.

The heterogeneity of the German news corpus may also explain another difference: while the final punctuation of the previous sentence (*FinPun*) is among the less useful features for English and Greek (albeit still outperforming the baseline), it is the best performing feature for German. The German news corpus contains many "sentences" that end in atypical end-of-sentence markers such as semi-colons (which are found often in cinema listings). Atypical markers will often not occur before paragraph breaks, whereas typical markers will. This fact renders final punctuation a better predictor of paragraph breaks in the German corpus than in the other two corpora.

The language models behave similarly across domains and languages. With the exception of the news domain, they do not seem to be able to outperform the majority baseline by more than 1%. The word entropy rate yields the worst performance, whereas character-based models perform as well as word-based models. In general, our results show that language modelling features are not particularly useful for this task.

4.2 The Influence of Syntactic Features

Our second set of experiments concentrated solely on the English data and investigated the usefulness of the syntactic features (see Table 3). Again, we created one-feature classifiers and a classifier that combined all features, i.e., language and character models, non-syntactic, and syntactic features ($all_{ns\_lcm\_syn}$). Table 3 also repeats the performance of the two baselines (B_d and B_m) and the combined non-syntactic models (all_{ns_lcm}). The accuracies of the three best performing one-feature models and the combined model are again shown in boldface.

As can be seen, syntactic features do not contribute very much to the overall performance. They only increase the accuracy by about 1%. A χ^2 test revealed that the difference between all_{ns_lcm} and $all_{ns\_lcm\_syn}$ is not statistically significant (indicated by $\not\dagger$ in Table 3) for any of the three domains.

	English		
feature	fiction	news	parl.
B_d	60.16	51.73	59.50
B_m	71.04	51.44	69.38
Cue_s	71.48	51.49	40.64
Cue_m	70.97	54.28	59.03
Cue_e	71.04	51.78	31.61
Parse	71.04	51.88	30.62
Num_s	71.04	53.56	69.05
Num_{vp}	71.04	54.18	70.59
Num_{np}	71.77	**56.11**	68.94
Num_{pp}	71.04	53.61	64.98
Num_{adjp}	71.04	51.11	42.62
Num_{advp}	71.04	52.40	47.96
Sign	**75.39**	**57.02**	67.95
$Sign_p$	**75.49**	**59.18**	**70.76**
$Childr_{s1}$	71.69	55.87	**79.35**
$Childr_s$	**75.34**	55.53	**79.52**
$Branch_s$	71.35	55.82	69.11
$Branch_{vp}$	71.33	53.46	70.48
$Branch_{np}$	71.77	**56.11**	33.09
$Branch_{pp}$	71.04	51.44	30.62
TreeDepth	72.57	54.04	69.00
all_{ns_lcm}	82.45	70.77	82.71
$all_{ns\_lcm\_syn}$	**82.91**$*\not\dagger$	**71.83**$*\not\dagger$	**83.92**$*\not\dagger$

Table 3: Syntactic features on English test data

The syntactic features seem to be less domain dependent than the non-syntactic ones. In general, the part-of-speech signature features (*Sign*, $Sign_p$) are a good predictor, followed by the syntactic labels of the children of the top nodes ($Childr_s$, $Childr_{s1}$). The number of NPs (Num_{np}) and their branching factor ($Branch_{np}$) are also good indicators for some domains, particularly the news domain. This is plausible since paragraph initial sentences in the Wall Street Journal often contain named entities, such as company names, which are parsed as flat NPs, i.e., have a relatively high branching factor.

4.3 The Effect of Training Size

Finally, we examined the effect of the size of the training data on the learner's classification accuracy. We conducted our experiments solely on the English data, however we expect the results to generalise to German and Greek. From each English training set we created ten progressively smaller data sets, the first being identical to the original set, the second containing 9/10 of sentences in the original train-

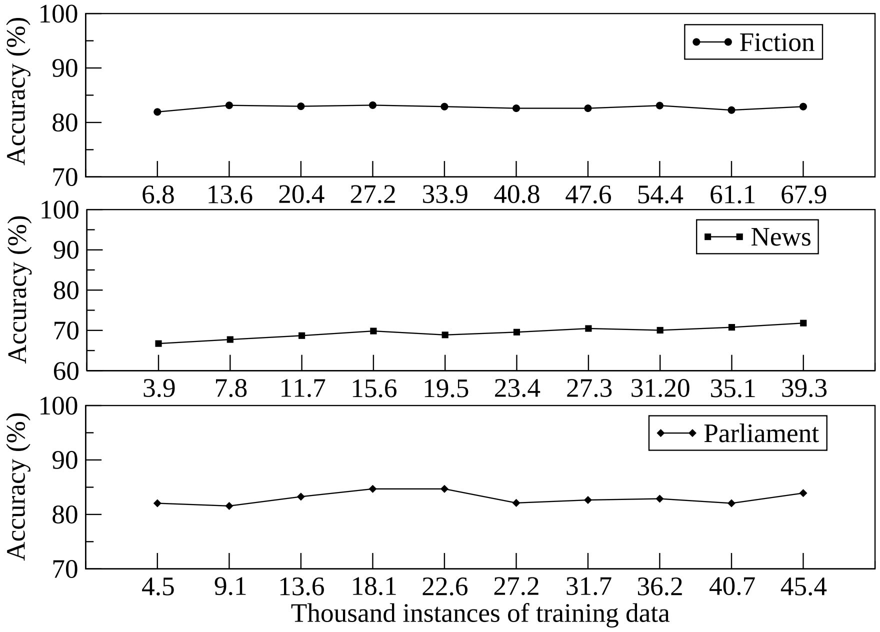

Figure 1: Learning Curves for English

	Kappa	% Agr
fiction	.72	88.58
news	.47	77.45
parl.	.76	88.50

Table 4: Human agreement on the paragraph identification task

ing set, the third containing 8/10, etc. The training instances in each data set were selected randomly. BoosTexter was trained on each of these sets (using all features), as described previously, and tested on the test set.

Figure 1 shows the learning curves obtained this way. The curves are more or less flat, i.e., increasing the amount of training data does not have a large effect on the performance of the model. Furthermore, even the smallest of our training sets is big enough to outperform the best baseline. Hence, it is possible to do well on this task even with less training data. This is important, given that for spoken texts, paragraph boundaries may have to be obtained by manual annotation. The learning curves indicate that relatively modest effort would be required to obtain training data were it not freely available.

4.4 Human Evaluation

We established an upper bound against which our automatic methods could be compared by conducting an experiment that assessed how well humans agree on identifying paragraph boundaries. Five participants were given three English texts (one from each domain), selected randomly from the test corpus. Each text consisted of approximately a tenth of the original test set (i.e., 200–400 sentences). The participants were asked to insert paragraph breaks wherever it seemed appropriate to them. No other instructions were given, as we wanted to see whether they could independently perform the task without any specific knowledge regarding the domains and their paragraphing conventions.

We measured the agreement of the judges using the Kappa coefficient (Siegel and Castellan, 1988) but also report percentage agreement to facilitate comparison with our models. In all cases, we compute pairwise agreements and report the mean. Our results are shown in Table 4.

As can be seen, participants tend to agree with each other on the task. The least agreement is observed for the news domain. This is somewhat expected as the Wall Street Journal texts are rather difficult to process for non-experts. Also remember, that our subjects were given no instructions or training. In all cases our models yield an accuracy lower than the human agreement. For the fiction domain the best model is 5.67% lower than the upper bound, for the news domain it is 5.62% and for the parliament domain it is 5.42% (see Tables 4 and 3).

5 Conclusion

In this paper, we investigated whether it is possible to predict paragraph boundaries automatically using a supervised approach which exploits textual, syntactic and discourse cues. We achieved accuracies between 71.83% and 83.92%. These were in all but one case significantly higher than the best baseline.

We conducted our study in three different domains and languages and found that the best features for the news and parliamentary proceedings domains are based on word co-occurrence, whereas features that exploit punctuation are better predictors for the fiction domain. Models which incorporate syntactic and discourse cue features do not lead to significant improvements over models that do not. This means that paragraph boundaries can be predicted by relying on low-level, language independent features. The task is therefore feasible even for languages for which parsers or cue word lists are not readily available.

We also experimented with training sets of different sizes and found that more training data does not necessarily lead to significantly better results and that it is possible to beat the best baseline comfortably even with a relatively small training set.

Finally, we examined how well humans do on this task. Our results indicate that humans achieve an average accuracy of about 77.45% to 88.58%, where some domains seem to be easier than others. Our models achieved accuracies of within 6% of human performance.

In the future, we plan to apply our model to new domains (e.g., broadcast news or scientific papers), to non-Indo-European languages such as Arabic and Chinese, and to machine generated texts.

References

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

Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 116–123, Toulouse.

Heidi Christensen, Bala Kolluru, Yoshi Gotoh, and Steve Renals. 2004. From text summarisation to style-specific summarisation for broadcast news. In *Proceedings of the European Conference on Information Retrieval*, Sunderland.

Philip Clarkson and Ronald Rosenfeld. 1997. Statistical language modeling. In *Proceedings of ESCA EuroSpeech'97*, pages 2707–2710, Rhodes.

Dmitriy Genzel and Eugene Charniak. 2003. Variation of entropy and parse trees of sentences as a function of the sentence number. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 65–72, Sapporo.

Alexander G. Hauptmann and Michael A. Smith. 1995. Text, speech and vision for video segmentation: The informedia project. In *Proceedings of the AAAI Fall Symposium, Computational Models for Integrating Language and Vision*, Cambridge, MA.

Marti A. Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Alistair Knott. 1996. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. Ph.D. thesis, Department of Artificial Intelligence, University of Edinburgh.

Philipp Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation. Unpublished Draft, `http://www.isi.edu/~koehn/publications/europarl.ps`.

R. E. Longacre. 1979. The paragraph as a grammatical unit. *Syntax and Semantics*, 12:115–134.

Fuchun Peng, Dale Schuurmans, Vlado Keselj, and Shaojun Wang. 2003. Language independent authorship attribution using character level language models. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 267–274, Budapest.

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

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, DC.

Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Sidney Siegel and N. John Castellan. 1988. *Non Parametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York.

Heather A. Stark. 1988. What do paragraph markings do? *Discourse Processes*, 11:275–303.

Mark Stevenson and Robert Gaizauskas. 2000. Experiments on sentence boundary detection. In *Proceedings of the 6th Applied Natural Langauge Processing Conference*, pages 84–89, Seattle, WA.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 491–498, Toulouse.

Chunshen Zhu. 1999. Ut once more: The sentence as the key functional unit of translation. *Meta*, 44(3):429–447.