Supervised classification of end-of-lines in clinical text with no manual annotation

Pierre Zweigenbaum
LIMSI, CNRS,
Université Paris-Saclay
91405 Orsay, France
pz@limsi.fr

Cyril Grouin
LIMSI, CNRS,
Université Paris-Saclay
91405 Orsay, France
grouin@limsi.fr

Thomas Lavergne
LIMSI, CNRS, Univ. Paris-Sud,
Université Paris-Saclay
91405 Orsay, France
lavergne@limsi.fr

Abstract

In some plain text documents, end-of-line marks may or may not mark the boundary of a text unit (e.g., of a paragraph). This vexing problem is likely to impact subsequent natural language processing components, but is seldom addressed in the literature. We propose a method which uses no manual annotation to classify whether end-of-lines must actually be seen as simple spaces (soft line breaks) or as true text unit boundaries. This method, which includes self-training and co-training steps based on token and line length features, achieves 0.943 F-measure on a corpus of short e-books with controlled format, F=0.904 on a random sample of 24 clinical texts with soft line breaks, and F=0.898 on a larger set of mixed clinical texts which may or may not contain soft line breaks, a fairly high value for a method with no manual annotation.

1 Introduction

Text segmentation is a low-level task which contributes to the higher-level information extraction tasks performed by natural language processing; for instance, Smith (2011, p. 5) states that "*If we build a language model on poorly segmented text, for instance, its predictive performance will suffer.*" Specifically, splitting a text into sentences, despite its looking like a largely solved problem, continues to raise nagging issues for some ill-formatted texts such as clinical texts (Miller et al., 2015). Most methods and software performing higher-level tasks (e.g., cTAKES (Savova et al., 2010) and others), such as part-of-speech tagging, syntactic parsing, entity and relation extraction, depend on low-level processes such as sentence segmentation. This paper focuses on a little-addressed, basic component in the NLP pipeline, which impacts sentence splitting and hence subsequent processes. This component may be seen as the determination of *paragraph boundaries*, or the *classification of end-of-lines*.

The problem can be described as follows. In some plain text documents, such as e-mail messages, text fields in databases, or PDF documents converted into text, the line break or end-of-line mark may or not play the role of a boundary marker for a text unit (a title, a paragraph, etc.) and hence may or not mark a sentence boundary. In some text documents the end-of-line mark is always a paragraph (or title) boundary, and no problem occurs: subsequent processes such as sentence splitting can be run within each paragraph. But in some text documents, an end-of-line mark may occur in the midst of a paragraph, typically to "wrap" paragraphs that exceed some set length: depending on the origin of the text and on input and formatting conditions, this may have been caused by an automatic process ('hard' line wrapping in some text editors) or by manual intervention of the typist. Often enough these originating conditions are not precisely known at the time these documents are submitted to natural language processing. Preprocessing must then address this situation and include some solution to the classification of end-of-line marks (henceforth noted <EOL>), i.e., to determine whether an <EOL> must be considered as an actual text unit boundary (henceforth <TUB>) or should be considered as standing for a simple space (<SP>), meaning that this line has incurred "paragraph wrapping" and should be considered together with (e.g., pasted to) the next line to form a larger text unit.

Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2016), pages 80–88, Osaka, Japan, December 12th 2016.

This situation is mentioned by some authors, e.g., by Miller et al. (2015) for the MIMIC II clinical texts, or by Zweigenbaum & Grouin (2014) for the i2b2/UTHealth 2014 NLP challenge documents (Stubbs et al., 2015). It is probably present in a much larger set of text collections, hence is likely to create some problems for many systems and teams working with these documents. While the impact of these problems still needs to be assessed precisely (see, e.g., (Zweigenbaum and Grouin, 2014) for limited examples), and may depend on the type of processing that follows, the number of situations where they are likely to occur warrants an investigation into general methods to address the task of classifying <EOL> marks.

End-of-line classification is therefore the topic of the present paper. We address it as a problem in itself, independently of its impact on subsequent tasks, and thus perform intrinsic evaluations of its performance; extrinsic evaluations, for instance through its impact on sentence splitting accuracy, are left for future work. Because <EOL> classification, although often needed, is only a small piece of preprocessing in a larger natural language processing pipeline whose adaptation to a given clinical task generally already requires some human annotation effort, a supervised method requiring more human annotation is not desirable. We therefore endeavored to investigate methods which require no human annotation to perform this task.

2 Related work

The problem of <EOL> classification seems to be little explored in natural language processing (NLP), and the section that Smith (2011) dedicates to segmentation does not mention it. Some NLP research (Sporleder and Lapata, 2006; Filippova and Strube, 2006) has addressed paragraph segmentation from a quite different perspective: given a text split into sentences, determine paragraph boundaries. However, they started from texts where sentence boundaries were given, and the input texts were assumed to be "clean" from the point of view of <EOL> marks (i.e., either sentence boundaries are deterministically marked by <EOL>s or by XML markup). A few papers on clinical NLP have recently addressed it and proposed methods based upon heuristics and knowledge about the usual format of the texts (Zweigenbaum and Grouin, 2014) or supervised machine learning (Miller et al., 2015).

Some document analysis research considers the notion of paragraph when converting a document into text by optical character recognition (Radakovic et al., 2013) or to reformat a text obtained from a PDF file (Fang et al., 2011). To perform these tasks they need to decide whether or not an <EOL> marks a paragraph boundary: this problem is similar to the one we address here. Radakovic et al. (2013) check whether a line starts with certain symbols (e.g., bullet points) or character case (uppercase or lowercase letters), ends with a number, together with other clues related to the number of words in the line, to its left and right indent size, to character font size, line coordinates in the page, distance between lines, and the presence of images. Fang et al. (2011) use information about the vertical and horizontal positioning of characters in the page, which are not available in plain text, together with paragraph indenting information. In contrast to (Radakovic et al., 2013), they do not use clues obtained from text content. To determine the reading order in a set of text objects (lines, paragraphs, etc.), Aiello et al. (2002) combine information on the spatial positioning of these objects and on the probability that the part-of-speech tag of the first word in the next object follows those of the last two words in the current object, according to a language model. This is the only reference we found in document analysis where a language model is used to help decide whether or not two lines must follow each other (which would mean, for us, be merged within the same paragraph).

We address a situation where we are given plain text but no information on the original layout of the page, such as spatial positioning of lines or characters, font size, actual left or right indent size of each line in its displayed form, which play a key role in document analysis methods. Text content, in contrast, is readily exploitable: language models based on the distribution of word features at the beginning and end of lines, as well as the distribution of line lengths in documents, can be used as cues.

Additionally, we aim to find methods which involve no human annotation. We shall see in the following section that a method consists in framing the situation as a supervised learning problem where each whitespace in a text (including spaces and <EOL>s) must be classified as a simple space (<SP>) or a text

Corpus (Gutenberg identifier)	Language	Documents	Paragraphs	Lines (wn)	Words
Around the World in 80 Days (103)	en	37	1679	6106	62,752
Around the World in 80 Days (2154)	en	37	1734	6887	65,510
Le Tour du Monde en 80 Jours (800)	fr	37	2053	6962	66,878
De la terre à la lune (38674)	fr	28	1449	5609	53,724
Da terra à lua (28341)	pt	28	1503	5869	57,487
Reis naar de Maan in 28 dagen… (27309)	du	47	1905	6915	65,671
Around the World in 80 Days (103+2154)	en	74	3413	12,993	128,262
Le Tour (800) + De la terre (38674)	fr	65	3502	12,571	120,602
i2b2/UTHealth 2014 training corpus	en	790	—	73,590	488,904
test subset	en	64	3554	5619	38,167

Table 1: Corpora. Lines are measured on the *wn* version of the documents, paragraphs on the *ln* version.

unit boundary (<TUB>), where part of the training examples (spaces) are positive and the rest (<EOL>) are unannotated. Nigam et al. (2000) show how text classification obtained by a Naive Bayes classifier can be improved by exploiting unannotated data on top of annotated data. This is how they train a classifier on texts whose class is known, then use it in a self-training fashion to compute the probabilities of all classes for each unannotated text. The additional information thus obtained allows them to re-train the classifier then to iterate until convergence, according to the expectation-maximization algorithm (EM). The method we propose below to train an <EOL> classifier is related to this principle, but does not need an initial human annotation. Elkan and Noto (2008) propose a non-iterative method for this purpose, but it assumes that the annotated examples are drawn randomly from the positive examples, which is not the case in our situation. Yet another path would consist in considering the <EOL> annotations as ambiguous (both <SP> and <TUB>) and in applying the methods of (Wisniewski et al., 2014). However, this would create a systematic dependency between these two classes in these annotations, a situation in which learning is not guaranteed (Bordes et al., 2010).

3 Material and methods

3.1 Corpora

We target here clinical texts with a complex mixture of formats. However, we also test our methods on more controlled corpora which we have in several formats. The controlled-format corpora are made of six plain text e-books by Jules Verne in four languages from the Gutenberg project (`http://www.gutenberg.net`), which we split into chapters. Each of their paragraphs is split into multiple lines (wrapped) if it exceeds a given threshold, and is bounded by blank lines. We consider each of these e-books, and the combination of the two English e-books and that of the two French e-books (see Table 1).

We produced versions of the e-books in which two properties of the paragraphs were set. First, a paragraph can be wrapped (*w*: broken into separate lines, the original format in this case) or typeset as one long line (*l*). Second, paragraph boundaries can be marked with a blank line (*b*) or not (*n*). This results in four combined formats. Among these, *wn* is the most difficult format to handle: it is the only one which has no simple paragraph delimitation. This is the one our system is meant to address. In *wb*, apart from obvious blank line separators, all end of lines should be classified as <SP>s: it is used to test whether our system produces false negatives. Conversely, *ln* has no obvious separators but no wrapped line at all: it is used to test whether our system produces false positives. Finally, *lb* has no wrapped line at all, and blank line separators everywhere, so our system handles it perfectly without classification.

The clinical text corpus is the i2b2/UTHealth 2014 NLP challenge (Stubbs et al., 2015) training corpus, which contains 790 records. These records keep the layout of a printed document and can include fixed-width columns, blank lines between text lines to reproduce double spacing, approximate positioning of elements on the page (tabulation, multiple spaces), table column separators represented by a special character (circumflex accent, pipe), etc. Some of the files have wrapped paragraphs, other do not.

We first preprocessed these texts to handle what is probably double line-spacing documents. To focus

on the paragraph wrapping problem, we handled double line-spacing deterministically by removing every other blank line in texts whose ratio of contiguous pairs of non-blank lines over the number of blank lines is below an empirical ceiling of 10%.

For evaluation during the development and test of our present <EOL> classifier, we manually annotated <EOL>s in a randomly sampled 64-document subset of the i2b2 training corpus. After initial annotation of another, smaller sample by two annotators and observation of a near-perfect inter-annotator agreement, we decided that this subset could be annotated by only one annotator. Note that since our method uses no manual annotation, the human annotations in these documents were not used to train the system, only to evaluate it. Specifically, using a plain text editor (emacs), we marked each <EOL> with a code as follows:

- 0 (no paragraph wrapping) means this <EOL> is a <TUB>.
- 1 (paragraph wrapping) means this <EOL> should be considered as an <SP>.
- 2 means there is no ambiguity in the present <EOL> (this is further explained below): it must be considered as a <TUB>, but no problem needs to be solved nor evaluated here.

3.2 Task modeling

We decompose the overall task of end-of-line classification into two parts:

1. Determine whether a document is subject to paragraph folding, i.e., whether it is liable to contain at least one <EOL> which should be categorized as an <SP>.
2. In a document which is subject to paragraph folding, classify <EOL>s as <SP> or <TUB>.

We focus here on the second part of the task, <EOL> classification proper, assuming that the first (easier) part, document classification, is solved, for instance in a supervised way. We also present a direction to address document classification with no human annotation, which we integrate into our general method.

3.3 Determination of documents with folded paragraphs

A text in which paragraphs are folded is likely to have a number of lines with similar lengths, which should thus be close to the mean line length in this text. Conversely, a text in which some lines are much longer than most other lines is probably not subject to paragraph folding, otherwise these longer lines would have been folded. The distribution of line lengths in a text, compared to their mean length, should therefore be a useful clue to determine whether a text is likely to have incurred paragraph folding.

Zweigenbaum and Grouin (2014) used the *coefficient of variation* of line lengths in a document (more detail is given below in the description of individual features). They set a threshold in a supervised way, over which a document was considered not to incur paragraph folding: documents with a larger variation in line lengths have a higher coefficient of variation, whereas documents with many lines of similar length have a smaller coefficient of variation. We use the same feature in the present work, but as a discretized value and with no manual annotation.

3.4 Blank line handling

We first solve an easy case which requires no further classification effort: a blank line, i.e., a line which is empty or only contains whitespace characters, such as whitespace and tabulations, is always considered as marking a paragraph boundary (as explained above, double line-spacing is removed if present). There is therefore no need for the classifier to learn to detect these lines: the <EOL> which ends a blank line, as well as the <EOL> of the preceding line, are both unambiguous. They are tagged with the "2" class in the training corpus and are excluded from the evaluation. These two <EOL>s remain useful however to train the classifier, because they participate in the estimation of the probabilities of occurrence of the preceding or following tokens given a <TUB> class.

3.5 End-of-line classification by self-training on noisy data and and co-training

The input to this task is made of tokenized texts, where punctuation has been separated from words. These texts contain whitespace spans which can play the role of simple spaces (<SP>) or text unit boundaries (<TUB>). We define the task as deciding, for each <EOL>, whether it should be considered as an <SP> (i.e., this is an <EOL> which is the result of paragraph folding and should thus be converted to a space) or a <TUB> (i.e., this is a true text unit boundary).

We address this task through a first subtask which consists in learning, *for each whitespace span* in a document (whether space or <EOL>), whether it should be classified as an <SP> or an <TUB>. The peculiarities of the training data and processes in this learning task are the following:

- <EOL>s are ambiguous: we do not know at this stage whether they are true <TUB>'s or actually <SP>'s. This results in a partial annotation of the corpus: spaces are unambiguously tagged as <SP>, but we do not know the actual tags for <EOL>s. We convert this situation into a *noisy annotation* by tagging every <EOL> with <TUB>: these tags are sometimes correct and sometimes incorrect.

- Because only <EOL> marks are ambiguous in our overall task, we are only interested in applying our classifier to them, not to spaces. Training is performed on these noisy annotations, using token-based features as described below, and learns a model M_A for tagging <EOL>s.

- Model M_A is applied to the <EOL>s in the training corpus itself, resulting in (noisily) disambiguated <EOL> tags (self-training).

- A second model M_B is learned on these new annotations, using different features (co-training). M_B is applied to the <EOL>s in the training corpus, resulting in modified <EOL> tags.

- Alternatively, a combined model $M_{A \cdot B}$ is built and applied to the <EOL>s in the training corpus.

We thus obtain, by exploiting only the naturally available information, three models and their associated <EOL> annotations. The process could be iterated, but we leave that for future work.

3.6 Features

We characterize a whitespace position s_i (space or <EOL>) between two words in a document d with the four discrete features below:

A_1 Left token: as in (Radakovic et al., 2013), we assume some tokens or punctuations are more often found at the end of a paragraph, while others are less often found there.

A_2 Right token ; similarly, we expect that some tokens or punctuations are often found at the start of a paragraph, or on the contrary are seldom found in this position.

A_3 Typographic form of the left token: all uppercase, capitalized, is a number, only contains punctuation (possibly differentiating between strong punctuations, i.e., period, exclamation mark and question mark, and the other punctuations), is a number followed by at least one punctuation and possibly preceded by one punctuation (typical form of bullet points). We assume that some typographic patterns, such as uppercase, are more frequent at the beginning of paragraphs, whereas others, such as strong punctuations, are more frequent at the end of paragraphs.

A_4 Typographic form of the right token.

These four features define a bigram language model, where a bigram is made of a whitespace position and an adjacent token or typographic form. Extension to longer n-grams is left for future work.

Additionally, we characterize a position s_i at the end of a line by the following two features:

B_1 l, length in characters of the line that ends with the space s_i, normalized (centered and reduced) as l_{norm} in document d (Equation 1a). We assume that a short line (or a very long line) is unlikely to need to be "pasted back" to the following line.

$$(a) \quad l_{norm} = \frac{l - \mu_d}{\sigma_d}, \qquad \mu_d = \frac{1}{N}\sum_{l \in d} l, \qquad \sigma_d = \sqrt{E[(l - \mu_d)^2]}, \qquad (b) \quad cv_d = \frac{\sigma_d}{\mu_d} \quad (1)$$

μ_d is the mean line length in document d and σ_d is the standard deviation of line lengths in d.

B_2 cv_d, coefficient of variation of line length l in document d (Equation 1b). This feature is common to all positions in document d. We assume that a document whose paragraphs are folded is likely to contain a number of lines of comparable lengths (probably close to the width of the input screen or original printable page). This should result in a low standard deviation of line length compared to the mean line length. The *coefficient of variation* cv is defined as their ratio (Equation 1b).

The latter two features l and cv_d have numeric values, we discretize them into ten bins between their minimal and maximal values as observed in the training corpus. If a test document has out-of-range values, they are discretized into the closest bin.

3.7 Naive Bayes classification

We use a very simple classifier, the *Naive Bayes* classifier, which is well-known for its robustness and speed. The probability of having a given class $c_j \in C$ (here, $C = \{$<TUB>, <SP>$\}$) for a certain whitespace s, characterized by features (e.g., $a_i \in A$), is (2a):

$$(a) \quad P(c_j|s) = \frac{P(c_j)P(s|c_j)}{P(s)}, \qquad (b) \quad P(s|c_j) = \prod_{i=1}^{|A|} P(a_i|c_j) \quad (2)$$

The Naive Bayes classifier hypothesizes the independence of the observed features for a given class. This leads to (2b). At inference time, the selected class is the one with the maximum a posteriori probability; since in (2a) $P(s)$ does not vary with the class c_j, we obtain (3a):

$$(a) \quad \arg\max_j P(c_j|s_i) = \arg\max_j P(c_j) \prod_{i=1}^{|A|} P(a_i|c_j), \qquad (b) \quad \hat{P}(a_i|c_j) \approx \frac{occ(a_i, c_j) + 1}{occ(c_j) + |C|} \quad (3)$$

Concretely, we compute the likelihood ratio of $c_{<\text{SP}>}$ over $c_{<\text{TUB}>}$, i.e., $\frac{P(c_{<\text{SP}>}|s_i)}{P(c_{<\text{TUB}>}|s_i)}$ and decide for an <SP> (tag 1) if greater than zero, <TUB> (tag 0) otherwise.

Equation (3a) relies on an estimation of $P(c_j)$ and $P(a_i)$ in a training corpus. This is classically performed according to a maximum likelihood principle: $\hat{P}(c_j) \approx \frac{occ(c_j)}{occ(s)}$ and $\hat{P}(a_i|c_j) \approx \frac{occ(a_i,c_j)}{occ(c_j)}$ where $occ(s)$ is the total number of spaces in the corpus, $occ(c_j)$ is the number of spaces with class c_j, and $occ(a_i, c_j)$ is the number of spaces with feature a_i and class c_j. A commonly encountered problem is the presence of test examples whose feature values have no occurrence in the training corpus. We address it with a widespread method, Laplace smoothing (Manning et al., 2008) (see Equation 3b above).

Training for model M_A is performed on all spaces with language model features $A_1 \ldots A_4$, whereas training for model M_B is performed only on <EOL>s with length features B_1B_2. A study of the scores of the two models on the development corpus shows that their distributions have similar dynamics, thus that a combination of the two can be considered. We create this combination by multiplying the likelihood ratios of M_A and M_B for the same space, hence its name $M_{A \cdot B}$. A more sophisticated combination might improve performance, but would require an annotated corpus to optimize its parameters.

4 Results

Since we use no human annotation, we trained our model on the whole set of texts in each sub-corpus and applied it to the same sub-corpus: each e-book of Section 3.1, and the 790 texts in the i2b2 corpus. For the i2b2 corpus, we only have gold annotations for a 64-text subset, on which performance is measured.

Corpus	Model	Acc	P	R	F	Acc	P	R	F
		six individual e-books				two merged e-books			
e-books: wb	wrap-none	0	—	0	0	0	—	0	0
e-books: wb	wrap-all	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
e-books: wb	M_A	0.733	1.000	0.733	0.846	0.812	1.000	0.812	0.895
e-books: wb	M_B	**0.949**	1.000	**0.949**	**0.973**	0.992	1.000	**0.992**	**0.996**
e-books: wb	$M_{A \cdot B}$	0.833	1.000	0.833	0.908	0.902	1.000	0.902	0.948
e-books: ln	wrap-none	1.000	—	—	—	1.000	—	—	—
e-books: ln	wrap-all	0	0	—	0	0	0	—	0
e-books: ln	M_A	0.996	—	—	—	0.995	—	—	—
e-books: ln	M_B	0.833	—	—	—	**1.000**	—	—	—
e-books: ln	$M_{A \cdot B}$	**0.999**	—	—	—	0.999	—	—	—
e-books: wn	wrap-none	0.262	—	0	0	0.264	—	0	0
e-books: wn	wrap-all	0.738	0.738	1.000	0.849	0.736	0.736	1.000	0.848
e-books: wn	M_A	0.802	**0.998**	0.733	0.845	0.861	**0.998**	0.812	0.895
e-books: wn	M_B	**0.917**	0.939	**0.949**	**0.943**	**0.932**	0.922	**0.992**	**0.955**
e-books: wn	$M_{A \cdot B}$	0.875	**0.998**	0.833	0.907	0.924	0.994	0.902	0.945
		+wrap				all			
i2b2	wrap-none	0.459	—	0.000	0.000	0.804	—	0.000	0.000
i2b2	wrap-all	0.541	0.541	1.000	0.702	0.196	0.196	1.000	0.328
i2b2	M_A	0.900	0.890	0.930	0.910	0.846	0.565	0.930	0.703
i2b2	M_B	0.897	0.905	0.904	0.904	**0.960**	**0.893**	0.904	**0.898**
i2b2	$M_{A \cdot B}$	**0.919**	**0.916**	**0.937**	**0.926**	0.903	0.685	**0.937**	0.791

Table 2: Experiments on multi-format e-books and on the i2b2 evaluation corpus (+wrap = only files with paragraph wrapping); Acc = accuracy, P = precision, R = recall, F = F-measure. Note that *e-books: ln* has no wrapped paragraph hence no positive space hence no true positive, therefore it has P=R=F=0.

We evaluate results by measuring the classical accuracy, precision, recall, and F-measure, for a task whose goal is to detect whether an <EOL> should be considered a <SP>. A true positive (TP) corresponds to an <EOL> which is correctly classified as a <SP>. A false positive (FP) is an <EOL> incorrectly classified as an <SP>. A false negative (FN) occurs when an <EOL> is incorrectly classified as a <TUB>. Accuracy, precision, recall, and F-measure stem from these definitions.

We provide two simple baselines: wrap-none considers that every <EOL> is a <TUB>, and wrap-all considers that every <EOL> is a <SP>. They enable us to show the 'lift' brought by M_A.

Table 2 shows evaluation results for two series of experiments: (i) on the e-book corpus, with wrapped paragraphs separated by blank lines (*wb*), long-line paragraphs with no separating blank line (*ln*), and wrapped paragraphs with no separating blank line (*wn*); and (ii) on the i2b2 corpus, on paragraph-folded documents (+wrap) then on the whole Test corpus (all).

5 Discussion

The first baseline method *wrap-none* always has null recall and F-measure by definition. It obtains perfect accuracy when no paragraph is wrapped (*ebooks:ln*), and sets a baseline with a fair accuracy on the mixed-type *i2b2:all* corpus. The second baseline method *wrap-all* has perfect or null performance on the artificial *ebooks:wb* and *ebooks:ln* corpora respectively. It sets a useful baseline for the highly wrapped *ebooks:wn* and *i2b2+wrap* corpora. However, for the more difficult, mixed *i2b2 all* corpus, it performs poorly. M_A and subsequent models outperform these baselines in F-measure and accuracy on the highly wrapped *ebooks:wn* corpus (except the F-measure of M_A on *ebooks:wn*, which is only on par with that of *wrap-all* and on the i2b2 corpus).

On e-books, the best recall in condition *wb* is obtained by model M_B. Its accuracy looks lower in condition *ln*, but this is due to its complete failure on one document (Le tour du monde (800)), which warrants further exploration; on all other documents it creates no false positive and hence outperforms the other two models. In condition *wn*, the more difficult situation, M_B obtains the best recall and F-measure, whereas M_A and $M_{A \cdot B}$ obtain near-perfect precision. The merged documents (103+2154 and 800+3874, right pane of Table 2) obtain better results than each individual document they contain. Globally, recall increases by 5–8pt while precision remains constant or slightly decreased, resulting in a 1–5pt increase in F-measure. Specifically, on the merged French documents, the failure of M_B which occurred on one

of them disappears. Training size is thus an important factor. Not shown for reasons of space: apart from one exception mentioned above, performance was similar on the four languages (English, French, Portuguese and Dutch). The format of the texts and the consistency of their vocabulary are probably more important than the language, and this set of languages does not exhibit a wide morphological variation. In summary, in each of these tests on one type of text with very regular wrapped text, the length-based model M_B, trained on the output of M_A, performs best.

Our evaluation on i2b2 documents is unfortunately not directly comparable to the F=0.965 of (Zweigenbaum and Grouin, 2014): they used a different subset of the i2b2 corpus which is not public. Besides, they implemented an extensive set of heuristics, whereas our method relies on no heuristic and aims at being more generic. Miller et al. (2015) used human annotations on a different corpus, and their results are thus not directly comparable to the present ones either.

The documents of the i2b2 evaluation corpus which incur paragraph wrapping display a more complex distribution of *w/l* and *b/n* formats, even within one document. The three models perform well, with a small advantage to the combined model $M_{A \cdot B}$. On the full i2b2 evaluation corpus, the length-based model M_B brings a substantive improvement over M_A and fares the best in terms of precision and F-measure. We hypothesize that this is due to the ability of its coefficient of variation feature to detect texts that are not likely to incur paragraph wrapping, thereby integrating a partial solution to this document classification subtask. In this setting the combined model $M_{A \cdot B}$ strongly improves the precision and F-measure of model M_A: the detection of non-paragraph-wrapping texts by the length-based features removes a large number of false positives and slightly improves recall at the same time.

In summary, if one wants robustness on both types of texts with a balanced precision, recall, and F-measure, the length-based model M_B, trained with annotations obtained from the token-based model M_A, is the most stable choice, at or above 0.90 for all these measures. Its features help the language-model model M_A reach a better precision and F-measure for the documents which are not subject to paragraph wrapping. However, if one looks for high-recall detection of wrapped paragraphs, model M_A has a higher recall of 0.93, and its combination with M_B was the most successful on wrapped texts. Therefore, depending on the needs of the natural language processing task that will be run on these texts, the choice of the model allows to favor precision (M_B) or recall (M_A or, better, $M_{A \cdot B}$). Besides, models (M_A or, better, $M_{A \cdot B}$) show their true performance on texts with paragraph wrapping. Finally, the combination $M_{A \cdot B}$, albeit efficient on i2b2 wrapped texts, was not sufficient to block false positives on texts with no wrapped paragraphs. Other strategies might be more successful, such as using M_B in a first step as a filter to detect and exclude non-wrapped documents, hence restricting the application of M_A or M_{AB} to an automatically detected +wrap subset.

A most interesting perspective is the study of the interaction of <EOL> classification with sentence segmentation. On the one hand, as suggested by one of the reviewers, sentence segmentation might be used as a baseline for <EOL> classification, all the more in texts where paragraphs typically end with a period. On the other hand, the study of the impact of <EOL> classification on sentence segmentation is one of the motivations for the present work, and constitutes our next step. As suggested by another reviewer, section title detection (Tepper et al., 2012) can also help paragraph segmentation. As a matter of fact, it was part of the heuristics used in (Zweigenbaum and Grouin, 2014), where it helped to avoid pasting a title (possibly with no final period) to the next line.

6 Conclusion

We presented a method which uses self-training and co-training to classify <EOL>s with no human annotation, based on available token and line length features. It achieves high <EOL> classification F-measures on i2b2 clinical texts which incur paragraph folding, and can also detect texts which are not subject to this phenomenon.

In future work, we plan to test M_B as a filter as outlined above. We will also explore other features such as POS tags and n-grams with $n > 1$, more powerful classifiers such as logistic regression and SVM, and perform extrinsic evaluations such as the impact on sentence segmentation.

Acknowledgements

This work was partially funded by BPI under FUI-15 grant SONAR and by the European Union's Horizon 2020 Marie Sklodowska Curie Innovative Training Networks—European Joint doctorate (ITN-EJD) under grant agreement No:676207, Methods in Research on Research (MiRoR).

References

Marco Aiello, Christof Monz, Leon Todoran, and Marcel Worring. 2002. Document understanding for a broad class of documents. *IJDAR*, 5:1–16.

Antoine Bordes, Nicolas Usunier, and Jason Weston. 2010. Label ranking under ambiguous supervision for learning semantic correspondences. In *27th International Conference on Machine Learning (ICML 2010)*, pages 103–110.

Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 213–220, New York, NY, USA. ACM.

Jing Fang, Zhi Tang, and Liangcai Gao. 2011. Reflowing-driven paragraph recognition for electronic books in PDF. In Gady Agam and Christian Viard-Gaudin, editors, *DRR*, volume 7874 of *SPIE Proceedings*, pages 1–10. SPIE.

Katja Filippova and Michael Strube. 2006. Using linguistically motivated features for paragraph boundary identification. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 267–274, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Timothy A Miller, Sean Finan, Dmitriy Dligach, and Guergana Savova. 2015. Robust sentence segmentation for clinical text. In *Proc AMIA Symp*, pages 112–113, San Francisco, Ca. AMIA.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, mai.

Bogdan Radakovic, Sasa Galic, and Aleksandar Uzelac. 2013. Paragraph recognition in an optical character recognition (OCR) process. United States Patent 8,565,474 B2, US Patent Office, octobre.

Guergana K. Savova, James J. Masanz, and Philip V. Ogren. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17:507–513.

Noah A. Smith. 2011. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Caroline Sporleder and Mirella Lapata. 2006. Broad coverage paragraph segmentation across languages and domains. *ACM Trans. Speech Lang. Process.*, 3(2):1–35, juillet.

Amber Stubbs, Christopher Kotfila, Hua Xu, and Özlem Uzuner. 2015. Identifying risk factors for heart disease over time: Overview of 2014 i2b2/UTHealth shared task Track 2. *J Biomed Inform*.

M. Tepper, D. Capurro, F. Xia, L. Vanderwende, and M. Yetisgen-Yildiz. 2012. Statistical section segmentation in free-text clinical records. In *LREC 2012, Eigth International Conference on Language Resources and Evaluation*, Istanbul, Turkey. ELRA.

Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. 2014. Cross-lingual part-of-speech tagging through ambiguous learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1779–1785, Doha, Qatar. Association for Computational Linguistics.

Pierre Zweigenbaum and Cyril Grouin. 2014. Reformatting clinical records based on global layout statistics. In Olivier Bodenreider, José Luis Oliveira, and Fabio Rinaldi, editors, *Proceedings 6th International Symposium for Semantic Mining in Biomedicine (SMBM 2014)*, pages 53–60, Aveiro. University of Aveiro.