Available online at www.sciencedirect.com

ScienceDirect

Speech Communication 00 (2016) 1–22

Speech Communication

www.elsevier.com/locate/procedia

Improving the Automatic Segmentation of Subtitles through Conditional Random Field

Aitor Álvarez[a], Carlos-D. Martínez-Hinarejos[b], Haritz Arzelus[a], Marina Balenciaga[a], Arantza del Pozo[a]

[a] *Human Speech and Language Technology Group, Vicomtech-IK4, San Sebastian, Spain*
[b] *Pattern Recognition and Human Language Technologies Research Center, Universitat Politècnica de València, Spain*

Abstract

Automatic segmentation of subtitles is a novel research field which has not been studied extensively to date. However, quality automatic subtitling is a real need for broadcasters which seek for automatic solutions given the demanding European audiovisual legislation. In this article, a method based on Conditional Random Field is presented to deal with the automatic subtitling segmentation. This is a continuation of a previous work in the field, which proposed a method based on Support Vector Machine classifier to generate possible candidates for breaks. For this study, two corpora in Basque and Spanish were used for experiments, and the performance of the current method was tested and compared with the previous solution and two rule-based systems through several evaluation metrics. Finally, an experiment with human evaluators was carried out with the aim of measuring the productivity gain in post-editing automatic subtitles generated with the new method presented.

Keywords: automatic subtitling, subtitle segmentation, pattern recognition, machine learning

1. Introduction

Subtitles have acquired great relevance within the audiovisual community during the last years, mainly after the adoption of the new audiovisual directives (Article 7 of the Audiovisual Media Services Directive[1]) of the European Parliament and of the Council in March of 2010. This legislation regulates the rights of people with a visual or hearing disability, and moved member states to take the necessary measures to guarantee that the services of audiovisual providers under their jurisdiction are gradually more and more accessible by means of sign-language, audio-description, easily menu navigation and subtitling.

Given the new audiovisual legislation, broadcasters and subtitling companies are seeking automatic solutions to be more productive than with the traditional manual subtitling. At the same time, disability organisations are pushing for both quantity and quality of subtitles, in order to not

[1] http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32010L0013\&from=EN

only increment the percentage of subtitling in the TV and the Internet, but also request quality subtitles. As a result, the demand of automatic solutions for quality subtitling has grown fast in the audiovisual community.

Several parameters take part in the definition of what the quality of subtitles is [1]. Apart from features related to subtitle layout, duration and text editing, subtitling segmentation is one of the most relevant, as it was demonstrated in [2], a study whose aim was to verify whether a correct text chunking in subtitles had an impact on both comprehension and reading speed using human evaluators. Even though important differences were not found in terms of comprehension, they demonstrated that a correct segmentation by phrase or by sentence significantly reduced the time needed to read subtitles. Furthermore, the strong need for proper segmentation is supported by the psycholinguistic literature on reading [3], where the consensual view is that subtitle lines should end at natural linguistic breaks to improve readability and reduce cognitive effort produced by poorly segmented text lines [4].

In this article, a new method based on probabilistic Conditional Random Field is applied to the field of automatic subtitling segmentation for Basque and Spanish languages. This work is a continuation of the previous research presented in [5], in which Support Vector Machine and Logistic Regression classifiers were employed for the subtitling segmentation task in the Basque language. In the present study, the same Basque corpus was used in order to compare the performance using the new classification method. In addition, the work has been extended to the Spanish language. It allowed us to confirm that the new classification method employed was valid for different types of corpora and languages. Given that the results obtained in [5] by the Support Vector Machine and Logistic Regression classifiers were very close due to its similar nature, in this work the performance of Support Vector Machine and Conditional Random Field were compared for both languages, leaving out the Logistic Regression classifier. Besides these statistical techniques, two rule-based methods were selected as baseline systems, such as the Chink-Chunk and Counting Character methods. Both rule-based methods were modified and adapted to the subtitle segmentation task by including additional information related to the maximum amount of characters allowed per line, speaker change and timing issues.

The results achieved proved that Conditional Random Fields outperformed clearly the Support Vector Machine based technique in terms of accuracy and computation time for both languages, whilst the rule-based Chink-Chunk and Counting Character methods obtained the worst results.

The article is structured as follows. Section 2 describes existing work on automatic subtitling and segmentation. Section 3 presents the rule-based baseline systems, whilst Section 4 looks at the Conditional Random Field method and how it fits the subtitle segmentation task. Section 5 describes the methodology we designed and implemented to build the new classification approach. Section 6 presents the experimental framework and the evaluation metrics. Section 7 summarizes the evaluation results and the performance comparison between the methods based on Conditional Random Field and Support Vector Machine. Section 8 shows the human performance results in segmentation correction for two options of obtaining draft segmentations. Finally, Section 9 draws conclusions and describes future work.

2. Related work in Subtitle Segmentation

Automatic segmentation of subtitles is a novel line of research which has not been studied extensively up to the present. To date, most of the automatic subtitling solutions have not been capable of generating syntactic and semantically coherent breaks for quality segmentation and, thus, segmentation is mainly performed considering the maximum number of characters allowed per line or through manual intervention.

The subtitle segmentation is similar to other segmentation techniques that are necessary for many Natural Language Processing tasks: Dialogue Act segmentation [6], sentence boundary detection for Text-to-Speech [7], or punctuation mark enriched speech recognition output [8]. These applications can be used to improve the source data for other tasks such as summarisation [9] or Machine Translation [10]. Most of these works employ lexical features such as the word sequence (obtained from the speech transcription or recognition) and Part-Of-Speech (POS) labels [7]; apart from that, signal features obtained from the speech (e.g., pause durations [11]), or prosodic features [12] are frequently employed to complement the data obtained from lexical features. It is usual to employ combination of these different features to obtain a more accurate result.

With respect to the models used for the segmentation, there is a wide variety of them applied for this task: Hidden Markov Models (HMM) [13], Hidden Event Language Models (HELM) [14], Maximum Entropy models [15], Neural Networks [6], or Conditional Random Field (CRF) [16]. Many works employ different models for different sets of features and combine the results [17, 16, 14] to obtain a more accurate segmentation. Finally, although majority of research has been done on English corpora, other languages have been used in the segmentation problem, like Japanese [18], German [19], or Portuguese [20], among others.

When looking at subtitle segmentation, few works have been carried out in the field of automatic segmentation, like the study presented in [5], where automatic subtitle segmentation was treated as a machine learning problem. In this previous work, Support Vector Machine and Logistic Regression classifiers were built over a Basque corpus consisting of TV cartoon programs and subtitles generated by professional subtitlers. To this end, subtitles with correct or incorrect segmentation were divided into two classes. Positive (correct) feature vectors were extracted from professionally-created subtitle data and contained the segmentation marks found in the corpus, whilst negative (incorrect) vectors were generated by manually inserting improper segmentation marks. Classifiers where then trained on balanced sets formed with these two types of vectors and employed for the segmentation task. The feature vectors were composed by 4 types of characteristics related to timing, number of characters, speaker change and a perplexity value given by a language model built over the training data. During decoding, an iterative algorithm was in charge of generating all the possible candidates for a break at each iteration. These candidates included sequences of consecutive words that did not exceed the maximum allowed length in characters before and after segmentation points. Feature vectors were then computed from these candidates and measured against the machine-learned classifiers and optimal candidates selected according to the obtained score. Similar performance was obtained for the two classifiers under evaluation. In the case of the SVM, it achieved a precision of 82.0% and a recall of 69.0%, with an average F1-score of 74.7%.

However, through this method, only the possible segmentation points were estimated, without distinguishing between different types of breaks. This implies considering line-breaks and subtitle-breaks, which is a critical information, to automatically generate the final subtitles correctly. It has to be noted that not all the subtitles have to have the same number of lines; there can be subtitles with just one line combined with others with two lines depending on the content and the segmentation rules. It is therefore critical to differentiate between line-breaks and subtitle-breaks. Finally, the computation time needed to generate all the candidates and select the optimal ones in the method presented in [5] was inefficient for a real application. Computing the iterative algorithm for one hour of content with 900 subtitles it took four hours of processing time on an Intel(R) Xeon(R) 2.00GHz and 32GB based server.

The rest of works in the literature regarding subtitle segmentation are focused on comparing the comprehension and reading speech in live-respoken subtitles segmented in a correctly and poorly manner [2], measuring the impact of arbitrary segmented subtitles on readers [4], and on studying the way line-breaking is commonly performed [21]. None of these three last works include technology to automatically create and segment subtitles.

3. Rule-based methods for Segmentation

This section details the rule-based Counting Character and Chink-Chunk methods adapted to the task of automatic subtitle segmentation.

3.1. Counting Character method

Nowadays, automatic segmentation is mainly performed considering only the maximum number of characters allowed per line or through manual intervention, since most of the automatic subtitling solutions have not been able to discriminate the natural pauses, syntactic and semantic information relevant for quality segmentation. This technique can be considered as the simplest way to perform segmentation, and it usually tends to increase up the post-editing effort widely to correct badly segmented subtitles [22].

3.2. Chink-Chunk algorithm

The Chink-Chunk algorithm is based on the POS information and it is basically focused on the distinction between content words (C), function words (F) and punctuation marks (P) to insert segmentation breaks. In Algorithm 1, a pseudo-code of the Chink-Chunk algorithm is presented.

```
if $POS\_previous = P$ then
    insert_break();
else if $POS\_previous = C$ and $POS\_next = F$ then
    insert_break();
else
    next_POS();
```

Algorithm 1: The Chink-Chunk algorithm

This rule-based method can be considered an evolution of the previously described Counting Character method, since it also considers the POS information and punctuation marks to predict possible segmentation breaks.

4. Conditional Random Field for Segmentation

Conditional Random Field (CRF) has been applied in different domains and applications, such as computer vision [23], bioinformatics [24], and specially in Natural Language Processing (NLP). In the NLP field, applications go from recognition and classification in text and speech [25, 26, 27] to segmentation and labelling of text [28, 29, 30]. These last applications inspired this work on the application of CRF to the subtitle segmentation problem.

Segmentation of subtitles can be seen as a label assignment to the sequence of words to be segmented, where the labels will basically indicate if a word pertains to the extreme (beginning or end) of a segmentation unit. Following a statistical approximation, the objective is obtaining the optimal assignment from a sequence of words. If the sequence of words is $W = w_1^n = w_1 w_2 \cdots w_n$, and the sequence of labels is $L = l_1^n = l_1 l_2 \cdots l_n$, the problem can be statistically stated as:

$$\hat{L} = \operatorname*{argmax}_{l_1^n} \Pr(l_1^n | w_1^n) \tag{1}$$

The problem can be solved by defining a model to estimate $\Pr(l_1^n|w_1^n)$ from training data (training process) and applying a searching algorithm on that model for a given sequence of words (decoding process). CRF offer an appropriate framework for modeling conditional probability between input-output sequences, as well as searching algorithms that allow to obtain the decoding results.

Following a notation similar to that employed in [31], a linear chain Conditional Random Field can be formulated as:

$$\Pr(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_{\tau=1}^{\mathcal{T}} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_\tau, y_{\tau-1}, \vec{x}_\tau) \right) \tag{2}$$

In Equation (2), the meaning of the different terms is the following:

- $\vec{x}$ and $\vec{y}$ represent input and output sequences, respectively (both of size $\mathcal{T}$).
- $Z(\vec{x})$ is a normalization factor in order to ensure a proper probability distribution.
- f_k (with $k = 1, \ldots, K$) is the set of features functions; these feature functions establish the correspondence between input and/or output elements, and they actually form the probability distribution; in this formulation, they are said to be *bigram* models, since output in time $\tau - 1$ ($y_{\tau-1}$) is related to output in time τ (y_τ).
- θ_k (with $k = 1, \ldots, K$) is the set of weights associated to each feature function f_k.

In the case of subtitle segmentation, input is a sequence of feature vectors derived from the sequence of words to be segmented, whereas output is a sequence of labels that represent for each word its situation inside the segmentation. Details on the specific input features and output labels are described in Section 5. Feature functions f_k and weights θ_k will be obtained in the training process.

According to this formulation, the final CRF model for subtitle segmentation can be stated as:

$$\hat{L} = \underset{l_1^n}{\operatorname{argmax}} \Pr(l_1^n | w_1^n) = \underset{l_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} \exp\left(\sum_{k=1}^{K} \theta_k f_k(l_i, l_{i-1}, \vec{w}_i)\right) \tag{3}$$

As it can be seen, the maximization allows to avoid the normalization term $Z(\vec{x})$. Notice that $\vec{w}_i$ is the feature vector derived from word w_i in the input.

5. Methodology

5.1. Important considerations in subtitles' segmentation

Automatic segmentation of subtitles has some particularities which have to be considered carefully. Firstly, there are several features that have to be taken into account at the same time. Apart from the text analysis, a correct segmentation of subtitles depends on other characteristics like (1) the amount of characters allowed per line, (2) timing issues related to long pauses and speech rhythm, (3) speaker changes, (4) the preceding and posterior words to select the most appropriate break type and point, and (5) the subtitle persistence on screen, which has a real impact in the readability. Moreover, it has to be noted that although there are some standard guidelines for a correct subtitling, such as Ofcom's Guidance on Standards for Subtitling[2]; BBC's Online Subtitling Editorial Guidelines[3]; and ESIST's Guidelines for Production and Layout of TV Subtitles[4], each subtitling company tends to have its own subtitling rules which may differ with each others in some specific points.

Secondly, besides looking upon the characteristics described above, the segmentation should be done including a syntactic analysis to create linguistically coherent breaks. It is the preferred and most adopted solution in the subtitling community and it follows from experiments and conclusions in psycholinguistic research, which show that readers analyze texts considering syntactic information [32], grouping words corresponding to syntactic phrases and clauses [33]. Therefore, with the aim of facilitating readability, subtitle lines should thus be split according to coherent linguistic breaks and considering the highest possible syntactic node as possible.

Finally, the demand of automatic solutions for subtitling comes from the need of tools to operate fast and provide quality results. Within an automatic subtitling solution which includes speech

[2] http://www.ofcom.org.uk/static/archive/itc/itc_publications/codes_guidance/standards_for_subtitling/subtitling_1.asp.html

[3] http://www.bbc.co.uk/guidelines/futuremedia/accessibility/subtitling_guides/online_sub_editorial_guidelines_vs1_1.pdf

[4] http://www.translationjournal.net/journal/04stndrd.htm

recognition technology, it is expected an output with well formatted and segmented subtitles, and few recognition mistakes. Besides, it should be executed in the shortest time as possible, requiring optimal solutions with high performance and low processing cost.

5.2. Rule-based methods' configuration

The Counting Character and Chink-Chunk based methods were adapted to the automatic segmentation of subtitles task by including some extra parameters. In the case of the Counting Character method, in addition to the maximum amount of characters allowed per line, the speaker change information was also employed to perform segmentation when it was available.

Regarding the Chink-Chunk method, apart from the initial segmentation breaks proposed by the algorithm described in Subsection 3.2, the final subtitles were composed considering these chunks, the maximum number of characters allowed per line, the speaker change (when available) and a timing feature. This last timing parameter was related to the time difference between consecutive subtitles in the training corpus and it corresponded to the average time difference between those couple of subtitles which were split without having a speaker change mark and with a time difference higher than 40 milliseconds (the minimum time difference between consecutive subtitles in both corpora). This average time was computed individually for each data set in Basque and Spanish, and it was considered as a fixed rule to insert a segmentation break in those cases in which this value was exceeded between two consecutive words.

5.3. Conditional Random Field's configuration

Before constructing a CRF graphical model for any application, a dependence structure has to be previously defined, which will be obeyed by the class labels given the observed data. This structure defines the transitions between the class labels at the graph node. In a Markov dependence structure, each class label and its corresponding feature vectors depend on the neighboring class labels and their features in the predefined neighborhood distance.

With the aim of defining a dependence structure for automatic segmentation through a CRF graphical model, and supposing that there are no more than two lines in a subtitle unit, eight class labels were created to define the function of each word within the subtitle, as listed below:

- B-SU *(Begin-Subtitle):* For each first word in subtitles.
- I-LI *(In-Line):* For each word in subtitle which is not the first or last word of a line or subtitle.
- E-LI *(End-Line):* For each word which represents the last word of a line which does not correspond to the end of a subtitle (e.g. last word of the first line in a subtitle with two lines).
- B-LI *(Begin-Line):* For each first word in a line that is not the first word in a subtitle (i.e., first word in second line for subtitles with two lines).
- E-SU *(End-Subtitle):* Each final word of a subtitle.
- BE-SU *(BeginSubtitle-EndSubtitle):* For words in an one-word subtitle.
- BS-EL *(BeginSubtitle-EndLine):* For words in the first one-word line for subtitles with two lines.
- BL-ES *(BeginLine-EndSubtitle):* For words in the second one-word line for subtitles with two lines.

00:00,166 - 00:05,333
Come here ,
Mum come here

Example 1: Subtitle example composed by 6 words and 2 lines.

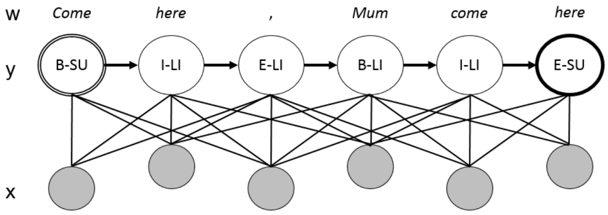

Fig. 1: Graphical model of the executed CRF over the Example 1. Transition factors depend on the surrounding two observations.

In Figure 1, a practical execution of the defined Markov dependence structure is presented, given the example subtitle shown in Example 1, which is composed by 6 tokens and formatted in 2 lines. Given this input example, the target of the CRF model would be to predict an output vector y = {y_1,y_2,..., y_N} through the observed feature vectors {x_1,x_2,..., x_N} extracted from the sequence of words {w_1,w_2,..., w_N}. In the CRF graphical models constructed for this work, each variable y_j corresponds to one of the class labels described above for each word at position j. For its part, each x_j contains the feature vector values about the word at position j. The transition factors of our CRF models depend on the surrounding two observations. The features used to describe each of the words at position j are composed of a total of 15 characteristics and they can be divided into the following subsets:

- Words: The current word and the surrounding 2 words. *(5 features)*
- Part-Of-Speech: The current word's Part-Of-Speech and the surrounding 2 words' Part-Of-Speech information. *(5 features)*
- Amount of characters per line and subtitle: A boolean value to control if the amount of characters per line and subtitle has been exceeded. The value was 0 while the accumulated amount of characters had not achieve the maximum quantity allowed per line and/or subtitle, or until there was a speaker change. Otherwise, the value was 1 for the current word. *(2 features)*
- Speaker Change: A boolean value to control if there is a speaker change in the current word or not. *(1 feature)*
- Time difference between the current and the neighboring words: Two parameters to compute the time difference between the current word and the previous and next word. We used 5 different discrete values for these parameters, including the value 0 for time differences lower than 100 milliseconds, 0.1 for differences between 100 and 500 milliseconds, 0.5 for differences between 500 and 1000 milliseconds, the value 1 for differences in the range of 1000 and 1500 milliseconds, the value 1.5 for differences higher than 1500 milliseconds and lower than 2000 milliseconds, and the value 2 for differences higher than 2000 milliseconds. The reference values were fixed looking at the training corpus, once all the time differences at the breaks were computed and analyzed. *(2 features)*

	Basque Corpus			Spanish Large Corpus			Spanish Small Corpora		
	CV	Training	Test	CV	Training	Test	CV	Training	Test
Programs	358	283	14	98	96	2	23	22	1
Subtitles	109006	86656	5307	81802	80058	1744	20154	19150	1004
Lines	166986	132832	8337	149774	146618	3156	37209	35356	1853
Words	768394	610471	37579	857648	839917	17731	211317	200964	10353
Lines/Subt	1.53	1.53	1.57	1.83	1.83	1.81	1.85	1.85	1.85
Words/Lines	4.60	4.60	4.51	5.73	5.73	5.62	5.68	5.68	5.59
Words/Subt	7.05	7.04	7.08	10.48	10.49	10.17	10.49	10.49	10.31

Table 1: Distinctive features of the different corpora (CV for cross-validation corpus, Training and Test for comparative corpus).

6. Experiments

6.1. Corpora description and processing

The four segmentation techniques under evaluation (CRF, SVM, Chink-Chunk, Counting Character) were tested over two languages, each with a particular corpus. For the Basque language, we used the same corpus of that employed in [5]. It was composed of TV cartoon programs in Basque with manually generated subtitles by professional subtitlers, for a total amount of 109,006 subtitles. The subtitle files were provided in SRT format, indicating start and end time-codes for each subtitle and presented in blocks of a maximum of two lines. The subtitles were carefully generated and segmented maintaining a linguistic coherence and splitting subtitles according to the highest possible syntactic node.

With regard to the Spanish language, the new corpus was composed of 98 episodes of the TV Spanish series ”Mi querido Klikowsky”, with a total amount of 81,802 subtitles. The subtitle files, which were provided also in SRT format, were created manually by professionals, and the segmentation was performed following specific predefined rules based on keeping a linguistic and syntactic coherence. The contents include many segments with spontaneous speech, grammatically incorrect sentences, and some words and expressions pronounced in several Spanish dialects, such as Argentinian and Andalusian. This issue triggers the Part-Of-Speech technology to make more mistakes than desired.

Since the Spanish 98 episodes do not include speaker change information, an additional subcorpora was also created to test the impact of this feature on the segmentation task for Spanish. We generated two additional subcorpora with 23 episodes from the original 98 ones, one containing speaker changes, which were included manually by a professional, and the other without speaker change information.

With regard to the feature vectors, the computation of the POS information was performed using the Eustagger toolkit [34] and ixa-pipe-pos [35] for the Basque and Spanish languages respectively. In addition, the time-codes at word level were obtained through the audio forced-alignment algorithms presented in [36] for both languages. This last information allowed us to obtain the differences in time between neighboring tokens.

Tables 1 and 2 describe the distinctive features and proportion of labels in all the corpora respectively.

6.2. Experiments setup

The Basque and Spanish CRF models were built and evaluated in two ways. Initially, the whole corpus for each language was used to train and evaluate models applying 10-fold cross validation technique. This evaluation was performed at class label and segmentation levels for both languages.

Each corpus was then split in train and test sets. This division was employed for comparing results with the ones obtained with the SVM based classification method and for assessing the

	Basque Corpus			Spanish Large Corpus			Spanish Small Corpora		
Label	CV	Training	Test	CV	Training	Test	CV	Training	Test
B-SU	14.06%	14.06%	13.99%	9.50%	9.50%	9.82%	9.53%	9.52%	9.68%
I-LI	56.69%	56.64%	55.79%	65.12%	65.13%	64.43%	64.80%	64.82%	64.22%
E-LI	7.47%	7.49%	7.97%	7.91%	7.91%	7.95%	8.06%	8.06%	8.18%
B-LI	7.52%	7.54%	8.03%	7.91%	7.91%	7.95%	8.06%	8.06%	8.20%
E-SU	14.11%	14.12%	14.05%	9.51%	9.50%	9.82%	9.53%	9.52%	9.70%
BE-SU	0.05%	0.05%	0.04%	0.02%	0.02%	0.01%	0.00%	0.00%	0.00%
BS-EL	0.08%	0.08%	0.10%	0.02%	0.02%	0.01%	0.01%	0.01%	0.02%
BL-ES	0.02%	0.02%	0.03%	0.01%	0.01%	0.01%	0.01%	0.01%	0.00%

Table 2: Proportion of the different labels in the different corpora (CV for cross-validation corpus, Training and Test for comparative corpus).

impact of the different features employed in the CRF models. In addition, the Chink-Chunk and Counting Character methods were also applied over the test sets and results were computed.

In the case of Basque, the division followed the partition made in [5] to evaluate the SVM based classification method. In this previous work, about 80% (86,656 subtitles) of the corpus was used to train the SVM models, 15% to evaluate them, and the rest (final-test) to evaluate the complete method including the iterative algorithm. For this work, we used the train and final-test partitions in order to compare both methods with the same size of corpus. Thus, 86,656 subtitles were used to train the Basque CRF models and 5,307 subtitles to test them. For the Spanish Large corpus without the speaker change information (98 episodes), the distribution was carried out keeping 80,058 subtitles for training and the rest (1,744 subtitles) for testing. Finally, for the two Spanish Small subcorpora (23 episodes) with and without speaker information, 19,150 of the subtitles were used to train models, and 1,004 subtitles for testing purposes.

The procedure followed to create segmentation breaks using the SVM based classification method was the same explained in the work [5], as it was briefly summarized in the previous Section 2. All the experiments related to CRF based models were performed using the CRF++ toolkit [37].

6.3. Evaluation metrics for segmentation

Apart from the classical metrics for label assignment (Precision, Recall, and F1-Score), since the problem to study was the subtitle segmentation, segmentation evaluation metrics had to be used. Four main evaluation metrics were used to test the performance of the developed segmentation techniques, as they are described in the following subsections.

6.3.1. F1-LINE

It is the evaluation metric proposed in [5] and it was only used in this work to compare the performance of the segmentation techniques in testing mode. It measures segmentation errors (false negatives and false positives) and correct segmentations (true negatives and true positives), and computes the accuracy through the F1-Score. It does not distinguish between line and subtitle breaks. The `conlleval` script[5] (which is the one used for the CoNLL-2000 shared task) was employed for measuring this metric, as well as for the Precision, Recall, and F1-Score calculations presented in Section 7.

6.3.2. NIST-SU

This well-known metric was provided by NIST for the Rich Transcription Fall evaluations [38], and it computes the number of segmentation errors (missed segments and false alarm segments) divided by the number of segments in the reference. Its limitation is that it does not consider

[5] `http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt`

position substitutions. For this work, it was computed at line level (NIST-SU-LI), which included both line-breaks and subtitle-breaks, and at subtitle level (NIST-SU-SUB).

6.3.3. DSER

It is computed dividing the number of incorrectly segmented portions in the reference by the total of segments in the reference. This is a more greedy metric if comparing with the NIST-SU, and its limitation lies in that it takes segments as whole sequence, and not as limits. For this work, it was computed at line level (DSER-LI), composed by line-breaks and subtitle-breaks, and at subtitle level (DSER-SUB).

6.3.4. SegER

It was proposed in [30] as an alternative evaluation measure to overcome the limitations posed by the previous NIST-SU and DSER metrics. SegER is computed as the edit distance between sequences of reference positions and hypothesis positions (those obtained automatically by the classifiers), using Insertion, Deletion, and Substitution as edition operations. As for the previous two metrics, it was also computed at line level (SegER-LI), which included both line-breaks and subtitle-breaks, and at subtitle level (SegER-SUB).

In Table 3 an example is given on how these metrics are computed taking as input the reference and the hypothesis, both composed of the class labels defined for the segmentation task. The computation scores of the segmentation measures are presented in Table 4.

Table 3: An example of how the different metrics are computed given a reference and the hypothesis estimated by the classifiers. For the *F1-LINE* metric calculation, *TN* means *True Negative*, *TP* corresponds to *True Positive*, *FP* is *False Positive* and *FN* means *False Negative*. The sign x corresponds to an error and ✓means correct. Finally, Correct and Substitution are represented by the C and S symbols respectively.

Segmentation measures											
Reference:	B-SU	I-LI	E-LI	B-LI	I-LI	E-SU	B-SU	E-LI	B-LI	I-LI	E-SU
Hypothesis:	B-SU	I-LI	E-LI	B-LI	E-SU	B-SU	I-LI	E-LI	B-LI	I-LI	E-SU
F1-LINE	*TN*	*TN*	*TP*	*TN*	*FP*	*FN*	*TN*	*TP*	*TN*	*TN*	*TP*
NIST-SU-SUB					x	x					✓
NIST-SU-LI			✓		x	x		✓			✓
DSER-SUB						x					x
DSER-LI			✓			x		x			✓
SegER-SUB					S_1	S_1					C
SegER-LI			C		S_1	S_1		C			C

Table 4: Computation scores of the example given in Table 3. It has to be noted that the scores can be positive (Acc, which means Accuracy) or negative (Err, which denotes Error).

Segmentation scores		
Metric	Computation	Score (Acc/Err)
F1-LINE	(2*TP) / (2*TP+FP+FN)	75% (Acc)
NIST-SU-SUB	2 Err / 3 Ref	66.67% (Err)
NIST-SU-LI	2 Err / 5 Ref	40% (Err)
DSER-SUB	2 Err / 2 Ref	100% (Err)
DSER-LI	2 Err / 4 Ref	50% (Err)
SegER-SUB	(1S) / (1C+1S)	50% (Err)
SegER-LI	(1S) / (3C+1S)	25 % (Err)

7. Results and discussion

7.1. Basque Corpus

7.1.1. Training and evaluation

This subsection describes the results obtained during the training and evaluation through the 10-fold cross-validation technique of the Basque CRF model using the whole corpus of this language. Table 5 presents the results for each class label, whilst Table 6 shows the accuracy reached and the number of tokens correctly tagged by the classifier.

Class labels evaluation				
	#	Precision	Recall	F1-Score
I-LI	435,575	93.5%	95.7%	94.6%
B-SU	108,014	87.3%	87.2%	87.2%
E-SU	108,441	87.4%	87.2%	87.3%
B-LI	57,819	69.3%	63.5%	66.3%
E-LI	57,392	69.0%	63.3%	66.1%
BL-ES	161	47.0%	5.0%	9.0%
BS-EL	589	90.1%	52.5%	66.3%
BE-SU	403	92.4%	84.9%	88.5%

Table 5: Precision, Recall and F1-Score values for each class label applying 10-fold cross-validation in the Basque corpus.

#Correct	679,347
#Labels	768,394
Accuracy	88.4%

Table 6: 10-fold cross-validation accuracy at class label level in the Basque corpus.

As it is shown in Table 5, among the most common labels in the Basque corpus, the labels representing the I-LI, B-SU, and E-SU labels reached the best results, obtaining a F1-Score of 94.6%, 87.2%, and 87.3% respectively. It means that the CRF classifiers modeled accurately subtitles boundaries and in-line words. However, the scores obtained at line-breaks level through the E-LI and B-LI labels are not as precise as at subtitle-breaks. It is due to the fact that there are more features at subtitle-level which could stand for a subtitle break, such as speaker changes, full stops or long silences, than at line-level, which usually depends exclusively on the syntactic information to predict a correct line break. Nevertheless, the F1-Score for the E-LI label achieved an interesting 66.1%. Besides, it has to be considered that the performance of the B-SU and B-LI labels are entirely dependent on the E-SU and E-LI labels respectively. However, as it can be appreciated in Table 6, if we consider the whole set of labels to be predicted (768,394 labels), an accuracy of 88.4% was achieved, given that 679,347 labels were correctly tagged.

Table 7: 10-fold cross-validation scores at segmentation level in the Basque corpus.

Segmentation evaluation					
NIST-SU-SUB	NIST-SU-LI	DSER-SUB	DSER-LI	SegER-SUB	SegER-LI
25.3	15.8	44.4	27.6	21.6	12.8

On the other hand, Table 7 presents the results for the NIST-SU, DSER and SegER evaluation metrics over the 10-fold cross-validation technique applied during the training of the CRF Basque model on the whole corpus. As it can be seen, the segmentation scores follow the same tendency as the example given in Table 4, where the segmentation errors at line-level are lower than at subtitle-level for any case. The CRF model achieved a promising performance for Basque. The interesting low error rates presented in Table 7 demonstrated the good performance of the labels, as it was shown in Table 5.

7.1.2. Testing and comparison

In this subsection, the Basque CRF model is compared at segmentation level with the Basque SVM based classification technique and the Chink-Chunk and Counting Character (CC) methods through the metrics described in Subsection 6.3, and using the train and test distributions described previously. In addition, the impact of discarding features from the CRF model was also evaluated.

Initially, Table 8 presents the Precision, Recall and F1-Score values achieved with the Basque CRF model over the Basque test data set. The CRF model was built on the train data set of the Basque corpus. Since the amount of the BL-ES, BS-EL and BE-SU labels was insignificant in the Basque test, we did not include their scores. As it can be seen in Table 8, the subtitles boundaries and inline words reached high accuracies, obtaining 90.6%, 86.6%, and 86.7% F1-Scores for I-LI, B-SU, and E-SU labels respectively. On the contrary, the performance of the labels related to the line boundaries was not as precise as the ones related to subtitle boundaries. The labels B-LI and E-LI, which correspond to begin-line and end-line words, achieved 44.4% and 44.5% F1-Scores values respectively. However, 31,200 of the 37,579 labels were correctly classified in overall, obtaining a global accuracy of 83.0%, as it is shown in Table 9.

Table 8: Precision, Recall and F1-Score values of each class label for the Basque test data set.

Class labels evaluation				
	#	Precision	Recall	F1-Score
I-LI	22,466	87.6%	93.9%	90.6%
B-SU	5,326	86.0%	87.2%	86.6%
E-SU	5,324	86.4%	87.1%	86.7%
B-LI	2,224	52.3%	38.5%	44.4%
E-LI	2,224	52.2%	38.7%	44.5%

Table 9: Accuracy at class label level for the Basque test data set.

#Correct	31,200
#Labels	37,579
Accuracy	83.0%

Table 10 presents the segmentation scores of each classification method for the Basque test set. As it can be seen, the low performance of the B-LI and E-LI labels presented in Table 8 had a real impact on the segmentation scores for the CRF model. For the NIST-SU and DSER metrics, the error rate at line level reached a higher error than the metrics related to the subtitle level. If we compare all the classification methods, the CRF model outperformed clearly the results obtained by the SVM-, Chink-Chunk- and CC-based classification methods for all cases. The difference is even higher for the metrics related to measure the subtitle boundaries.

Table 10: Segmentation scores of the CRF-, SVM-, Chink-Chunk- and CC-based methods for the Basque test set.

	Segmentation score						
	F1-LINE	NIST-SU-SUB	NIST-SU-LI	DSER-SUB	DSER-LI	SegER-SUB	SegER-LI
CRF	83.0	26.5	28.3	47.1	47.4	22.6	21.6
SVM	74.7	81.6	56.1	110.5	79.1	59.0	33.7
Chink-Chunk	56.6	108.3	87.3	129.7	116.7	69.1	53.9
CC	36.2	120.4	174.9	136.4	132.6	83.2	70.6

Table 11 presents the comparison of the results obtained with CRF for the Basque test set employing different sets of features: all 15 features (the same than those of Table 10), excluding the speaker change feature, excluding the POS features, excluding the time difference features, and excluding all at the same time (i.e., only word and characters per line/subtitle features).

Table 11: Segmentation scores of the CRF model for the Basque test set with different sets of features.

	Segmentation score						
CRF features	F1-LINE	NIST-SU-SUB	NIST-SU-LI	DSER-SUB	DSER-LI	SegER-SUB	SegER-LI
All	83.0	26.5	28.3	47.1	47.4	22.6	21.6
No speaker change	82.6	25.8	30.1	45.6	49.5	22.1	22.3
No POS	82.3	25.9	30.7	46.5	51.2	22.6	23.1
No time differences	79.6	41.3	31.2	70.4	51.4	33.3	23.3
Only word and characters	78.0	42.3	37.2	72.1	59.8	34.6	26.7

As it can be seen from these results, only the features related to time differences present by themselves a clear impact in the general performance of the label assignment (F1-LINE measure) and in the subtitle segmentation performance. The additional missing of the speaker change and

POS features produces a higher degradation of the results. In contrast, in the line segmentation performance the single impact of time difference features is similar to the rest of features, and only when the rest are missing there is a clear degradation in the performance. This can be intuitively explained by the fact that subtitles boundaries are sensitive to speech pauses (i.e., it is possible that between the end of a subtitle and the beginning of a next one there is a silence in the speech signal); meanwhile, general line breaks are not usually related to speech pauses, since many of them are caused by the necessity of splitting the subtitle into two lines, where there is no silence but a continuous voice signal. Moreover, it can be appreciated that the interaction among the different excluded features is the one that provides a beneficial line segmentation, since only when all of them are missing there is a substantial decrease of performance.

Another fact that can be deducted from these results is that, even with less parameters (which causes a degradation in performance) CRF models outperform the SVM-based technique and the rule-based Chink-Chunk and CC methods for the subtitle segmentation task.

7.2. Spanish Large Corpus

7.2.1. Training and evaluation

The results obtained during the training and evaluation of Spanish CRF models with the Spanish Large Corpus (98 episodes) and applying 10-fold cross-validation technique are presented in this subsection. This corpus did not include information about speaker changes. Table 12 describes the results at class label, and the accuracy along with the number of correctly tagged labels are shown in Table 13. Finally, the results at segmentation level are presented in Table 14.

Table 12: Precision, Recall and F1-Score values for each class label applying 10-fold cross-validation in the Spanish Large corpus (without speaker change information).

Class labels evaluation				
	#	Precision	Recall	F1-Score
I-LI	558,500	90.2%	92.6%	91.4%
B-SU	81,513	98.8%	93.6%	96.1%
E-SU	81,543	98.9%	93.6%	96.2%
B-LI	67,861	60.5%	57.7%	59.1%
E-LI	67,831	60.5%	57.8%	59.1%
BL-ES	111	75.0%	24.3%	36.7%
BS-EL	141	25.0%	0.7%	1.4%
BE-SU	148	100.0%	92.6%	96.1%

Table 13: 10-fold cross-validation accuracy at class label level in the Spanish Large corpus (without speaker change information).

#Correct	748,270
#Labels	857,648
Accuracy	87.3%

Table 14: 10-fold cross-validation scores at segmentation level in the Spanish Large corpus (without speaker change information).

Segmentation evaluation					
NIST-SU-SUB	NIST-SU-LI	DSER-SUB	DSER-LI	SegER-SUB	SegER-LI
7.4	34.6	11.8	57.9	7.2	25.9

In the case of the Spanish Large Corpus evaluation, the low accuracy of the labels at line level particularly affects the segmentation scores for all the metrics. As it can be seen in Table 14, the error rates of the metrics related to line boundaries are specially higher than the rates of subtitle boundaries. It can be explained by the really good performance of the B-SU, E-SU, and BE-SU labels involved in specifying the subtitles boundaries, achieving F1-Score of 96.1%, 96.2%, and 96.1% respectively. On the contrary, the B-LI, E-LI, and BS-EL labels scored accuracies of 59.1%, 59.1%, and 1.4% respectively.

7.2.2. Testing and comparison

In this subsection, the Spanish CRF model and SVM based classification technique, both built over the Spanish Large Corpus, which does not contain speaker changes, are compared at segmentation level. The results obtained when applying the Chink-Chunk and CC-based methods to the test set are also presented. Firstly, the results reached at label level through the CRF model are shown in Table 15 and Table 16. In this case, the impact of not having speaker changes marks is clearly appreciated in all the labels related to describe the subtitles and lines boundaries. The performance of B-SU and E-SU labels at subtitle level has decreased clearly when comparing with the Basque corpus. Besides, the precision of the B-LI and E-LI labels does not reach 45%. The only label which has kept a good performance is the I-LI label, achieving a F1-Score value of 90.4%.

Table 15: Precision, Recall and F1-Score values for each class label for the Spanish Large corpus test set.

Class labels evaluation				
	#	Precision	Recall	F1-Score
I-LI	11,966	88.3%	92.5%	90.4%
B-SU	1,094	98.2%	61.7%	75.7%
E-SU	1,093	98.2%	61.7%	75.7%
B-LI	1,788	44.7%	56.7%	50.0%
E-LI	1,789	44.8%	56.8%	50.1%

Table 16: Accuracy at class label level for the Spanish Large corpus test set.

#Correct	14,316
#Labels	17,731
Accuracy	80.7%

The results obtained at segmentation level over the test data set of the Spanish Large corpus are presented in Table 17 for all the methods under evaluation. Naturally, the lower performance of the previously described labels should affect directly all the metrics which measured segmentation of the CRF model. However, the differences at line level are not very low when comparing with the results obtained in the training and evaluation phase, where the performance of the B-SU/E-SU and B-LI/E-LI labels was better. Even though, the error rates grew notably at subtitle level if we compared with the results in Table 14.

Table 17: Segmentation scores of CRF-, SVM-, Chink-Chunk- and CC-based methods for the Spanish Large corpus test set.

	Segmentation score						
	F1-LINE	NIST-SU-SUB	NIST-SU-LI	DSER-SUB	DSER-LI	SegER-SUB	SegER-LI
CRF	80.7	39.4	38.2	58.0	61.6	38.8	28.4
SVM	41.4	146.6	119.2	159.4	140.0	82.7	66.6
Chink-Chunk	36.2	128.1	107.6	143.1	130.8	82.9	69.5
CC	15.2	142.9	136.2	148.9	148.1	91.1	87.8

As in Table 10, both the accuracy of F1-LINE and the error rates of the other metrics are outperformed by the CRF model when comparing with the other methods, which obtained error rates higher than the 100% for the NIST-SU and DSER metrics. The main reason for these high error rates is that these both methods generate break candidates, without distinguishing between line and subtitle breaks. Using these break points proposed, we assigned automatically labels to each word of the test contents, generating two-lines subtitles consecutively from the beginning of each content. This was the only way to create subtitles using the SVM-, Chink-Chunk- and CC-based method's outputs, since no more information was provided by these methods. This procedure could therefore generate multiple errors in tagging words with incorrect labels, and mainly in differentiating between the E-LI and E-SU labels.

In addition, it has to be considered that the Spanish Large corpus contains multiple segments with spontaneous speech, unfinished sentences and words, and expressions from Spanish dialects such as Andalusian and Argentine. One of the main parameters in the SVM based classification method, which is described in detail in [5], was the perplexity given by a language model (LM) built on the train data and using Part-Of-Speech (POS) tags as units. The difficulties posed by these type of contents produced mistakes in the POS information extraction, and thus in the high perplexities given by the LM. This also affected segmentation error rates to be extremely high for the SVM based classification method.

As it was done for the Basque corpus in Subsubsection 7.1.2, the impact of the features used in CRF models was assessed for this corpus. In this case, the complete set of features does not include the speaker change data; thus, the different subsets of features are: all the 14 features (the same than in Table 17), excluding the POS features, excluding the time difference features, and excluding both features (only words and characters features).

Table 18: Segmentation scores of CRF models for the Spanish Large corpus test data for different sets of features.

	Segmentation score						
CRF features	F1-LINE	NIST-SU-SUB	NIST-SU-LI	DSER-SUB	DSER-LI	SegER-SUB	SegER-LI
All	80.7	39.4	38.2	58.0	61.6	38.8	28.4
No POS	80.7	39.7	38.8	58.4	62.9	39.0	29.5
No time differences	75.6	79.7	40.0	118.1	63.9	56.7	29.2
Only words and characters	75.6	79.6	40.5	117.4	64.7	56.7	30.1

The results for this corpus reveal a similar behavior to that in the Basque corpus. The POS measures by themselves have a small impact, and the time difference measures have a high impact in the subtitle segmentation, but not in the general line segmentation. In this case, since the average number of lines per subtitle is higher than in the Basque corpus, the relative impact in the line segmentation is even lower. The absence of the two sets of features (both POS and time difference features) presents in general results similar to those of the set without time differences. As happened with the Basque corpus, in all cases CRF models still outperforms the SVM-, Chink-Chunk- and CC-based methods.

7.3. Spanish Small Subcorpora

7.3.1. Training and evaluation

The results reached during the training and evaluation of Spanish CRF models for the two subcorpora (23 episodes) with and without speaker change information and applying 10-fold cross-validation technique are presented in this subsection. This evaluation was focused on checking the impact of having speaker change information in the accuracy of the Spanish subtitle segmentation. Table 19 describes the results at class label, and the accuracy along with the number of correctly tagged labels are shown in Table 20. The BL-ES, BS-EL and BE-SU labels are not shown because of their low count. Finally, the results at segmentation level are presented in Table 21.

Table 19: Precision, Recall and F1-Score values for each class label applying 10-fold cross-validation in the Spanish Small corpora, with and without speaker change information.

		Class labels evaluation					
		With Speaker Change			Without Speaker Change		
	#	Precision	Recall	F1-Score	Precision	Recall	F1-Score
I-LI	136,933	89.9%	93.4%	91.6%	89.0%	92.4%	90.7%
B-SU	20,135	85.7%	71.6%	78.0%	85.4%	70.9%	77.5%
E-SU	20,137	85.8%	71.6%	78.0%	85.4%	70.9%	77.5%
B-LI	17,040	55.2%	57.3%	56.2%	53.0%	55.6%	54.3%
E-LI	17,038	55.2%	57.3%	56.2%	53.0%	55.6%	54.3%

Although the hypothesis was that the speaker change information should help improving the results, this issue was not clearly demonstrated for the Spanish Small corpus in 10-fold cross-validation technique. As it can be seen in Table 19, the differences of the B-SU and E-SU labels in terms of F1-Score are minimum between the two corpora, with and without speaker changes. The improvement is only 0.5 percentage points. The labels which represent the lines boundaries have a similar behavior, achieving improvements of almost 2 percentage points on average. These

Table 20: 10-fold cross-validation accuracy at class label level in the Spanish Small corpora, with and without speaker change information.

	With speaker change	Without speaker change
#Correct	176,301	174,025
#Labels	211,317	
Accuracy	83.4%	82.3%

small improvements are also present in Table 21. Even if all the error rates for the contents with speaker change were lower, the differences with the contents without speaker changes are not as clear as expected. The main reason could be related to the small size of the corpus used for these experiments. As it was described in Table 1, the Spanish Small Corpus was composed by a total amount of 20,154 subtitles, which corresponds to a quarter of the Spanish Large Corpus.

Table 21: 10-fold cross-validation scores at segmentation level in the Spanish Small corpora, with and without speaker change information.

Speaker	Segmentation evaluation					
change	NIST-SU-SUB	NIST-SU-LI	DSER-SUB	DSER-LI	SegER-SUB	SegER-LI
With	40.3	32.5	64.2	54.4	33.8	25.3
Without	41.2	36.1	65.4	60.0	34.5	28.0

7.3.2. Testing and comparison

In this last subsection, the CRF model and the SVM based classification method are compared for the two Spanish Small Corpora. Tables 22 and 23 present the scores for each label, whilst Table 24 shows the segmentation score and error rates for each corpus and classification method.

Table 22: Precision, Recall and F1-Score values for the Spanish Small corpora test set, with and without speaker change information.

		Class labels evaluation					
		With Speaker Change			Without Speaker Change		
	#	Precision	Recall	F1-Score	Precision	Recall	F1-Score
I-LI	6,833	89.9%	92.4%	91.1%	88.6%	90.6%	89.6%
B-SU	818	86.1%	70.3%	77.4%	84.0%	69.2%	75.9%
E-SU	818	86.3%	70.3%	77.5%	84.2%	69.2%	76.0%
B-LI	942	52.4%	58.2%	55.2%	48.9%	54.8%	51.7%
E-LI	942	52.4%	58.3%	55.2%	48.9%	54.9%	51.7%

The differences between the scores obtained with and without speaker information using separated train and test partitions are more significant than when the 10-fold cross-validation technique was applied. The improvements at subtitle and line levels are around 1.5% and 4% percentage points respectively in Table 22. The better performance of the CRF method against the SVM based classification method is demonstrated again in Table 24 for all the metrics. It is interesting to observe how the results of the SVM based classification method are better in this case comparing with the rates obtained with the whole Spanish corpus given in Table 17. For instance, the F1-LINE metric scores 45.4% of accuracy over the Spanish Small Corpus which does not include speaker changes, whilst an accuracy of 41.4% was obtained on the Spanish Large Corpus. The same tendency is kept for the rest of metrics. It can be explained by the fact that in the SVM based classification method the labels are almost randomly assigned just following the break marks given in the output. Hence, it seems that this method is prone to generate more errors as more subtitles are given to test.

On the contrary, the results of the CRF model are more consistent with the size of the corpus used to train and test models. In comparison with the Table 17, the metrics achieved higher error rates for the Spanish Small Corpora. Finally, the impact of the speaker change parameter

Table 23: Test set accuracy at class label level in the Spanish Small corpora, with and without speaker change information.

	With speaker change	Without speaker change
#Correct	8,539	8,342
#Labels	10,353	
Accuracy	82.5%	80.6%

Table 24: Segmentation scores of CRF and SVM models for the Spanish test data, including or not speaker change information

Speaker change	Model	Segmentation score						
		F1-LINE	NIST-SU-SUB	NIST-SU-LI	DSER-SUB	DSER-LI	SegER-SUB	SegER-LI
With	CRF	82.5	40.8	33.2	64.5	56.2	34.4	25.8
	SVM	47.5	115.8	90.8	136.6	119.7	74.2	59.5
Without	CRF	80.6	43.7	38.5	68.2	64.2	36.2	29.6
	SVM	45.4	118.0	93.6	138.6	120.0	76.1	61.1

is demonstrated in Table 24. Although the experiments were carried out with a small corpus, the error rates were lower for all the metrics in the corpus with speaker changes. The higher difference is given by the DSER-LI metric with a difference of 8 percentage points, reaching an error rate of 56.2% and 64.2% for the corpus with and without speaker changes respectively.

Since the effect of discarding other features was tested previously (see Subsubsections 7.1.2 and 7.2.2) and showed a consistent behavior, the equivalent experiment was not conducted for these corpora.

7.4. General discussion

Comparing all results from a general point of view, the first remarkable issue is that using CRF for assigning the different subtitle labels to the words is a valid alternative, since in all cases average accuracy is higher than 80%. Comparing with the previous SVM approximation employed in [5], it supposes a large impact on subtitling quality, specially for Spanish language (where SVM results present an accuracy lower than 50%), although Basque language presents a significant improvement as well.

Regarding the two rule-based baseline systems presented, the Chink-Chunk based method was carefully adapted to the subtitle segmentation task by using information related to the maximum amount of characters per line, speaker changes, and the time difference feature, which was selected as one of the most important parameter for both languages when using CRF models (Tables 11 and 18). Nevertheless, as in the case of the CC-based method, the results show a worse performance mainly because of the absence of a model built with training data and adapted to the characteristics of the domain.

Examining the results, the general tendency is having a more accurate subtitle segmentation than in-line segmentation. This is reasonable since begin and end of subtitles present more specific clues to detect its presence (e.g., punctuation marks, silences, speaker changes, etc.) than line breaks. When looking at the whole subtitle segmentation with respect to line segmentation (i.e., the one that includes all lines as units, independently if they are starting or end lines for subtitles), the general tendency is that line segmentation presents lower error than subtitle segmentation, which is reasonable since subtitle boundaries are a subset of line boundaries, and subtitle segmentation accuracy affects line segmentation accuracy.

However, in a few cases (Basque comparison, Table 10, and Spanish Large comparison, Table 17), differences show an irregular behavior, and even in the Spanish Large Corpus cross-validation experiments (Table 14), the tendency is the opposite. This can be explained by the nature of the corpora and the behavior of the classifier: Spanish Large Corpus presents a high proportion of lines in each subtitle (around 1.8 lines per subtitle, in contrast to what occurs with the Basque corpus with around 1.5 lines per subtitle), and presents a much lower relative accuracy of line boundaries labels (B-LI and E-LI) than the other cases (relative F1-Measure difference is about 60%, in contrast to about 30% in the Basque cross-validation and about 40% in the Spanish Small cross-validation). The combination of the two factors (higher number of lines and lower accuracy for detecting line boundaries with respect to subtitle boundaries) explains the different behavior, since there are more line boundaries to detect and they are detected with less precision, making the whole line segmentation error higher than the simple subtitle segmentation error. Similar arguments explain the irregular behavior of Basque comparison (less lines per subtitle but much lower detection of line boundaries) and Spanish Large comparison (same number of lines per subtitle but not so low performance on the detection of line boundaries).

These issues allow us to suppose that, given the nature of the corpus (specially proportion of lines by subtitle) and the classifier (accuracy in detecting line boundaries with respect to subtitle boundaries), different performances can be expected at the two levels (subtitle- and line-level) and decisions can be taken on the use of more specialized models for the nature of the corpus, which will allow to obtain more accurate results for the subtitle segmentation task.

In any case, CRF represents a new milestone in this task since results are in all cases much better than the current statistical alternative (SVM based classification method) and the decoding time is really fast (less than 0.1 milliseconds per subtitle in an Intel(R) Core(TM) i7 computer at 3.4 GHz with 16 GB of RAM) with respect to that provided by the SVM based classification method (16 seconds per subtitle on average).

8. Productivity gain evaluation

With the aim of testing the efficiency of the CRF classifier, an experiment was carried out with human evaluators. The experiment consisted of measuring the effort of post-editing the segmentation of subtitles generated using two techniques: (1) subtitles segmented using the CC-based method, and (2) those segmented using the information provided by the CRF classifier. Results from both techniques were finally compared to evaluate whether using the CRF-based classification method was more productive and facilitates the process of generating quality subtitles.

Nine students of the Subtitling Module included in the UAB's (Universitat Autònoma de Barcelona) METAV[6] and MTAV[7] Masters Programs volunteered to participate in the evaluation. In addition to the subtitling practice acquired through the masters program, they all had further subtitling expertise varying from one month to three years.

The experiment was performed over the Spanish corpus containing the information related to speaker change, composed of a total amount of 20,154 subtitles, 1,004 of which were used for testing purposes. This test set was first divided into smaller sets of 50 subtitles each, which were generated using both the CRF-based classification method and the CC-based method. Each participant was then asked to post-edit the segmentation of two sets, each of which had been segmented using one of the two techniques. In order not to influence the post-editing task, the evaluation sets assigned to each post-editor contained different subtitles. The participants received some previous guidelines on the manner they had to post-edit and correct the subtitles, including some specific and reference rules for a proper segmentation. Subtitling Workshop[8] and the Toggl[9] tools were employed as subtitling and time tracking software, respectively. After finishing the task, participants generated a Toggl report including the time required to complete it.

Figure 2 shows the time in minutes per subtitle (mps) needed by each participant to post-edit the 50 subtitles in the two sets segmented with the described two methods.

As it can be seen in Figure 2, all post-editors needed more time to post-edit a subtitle in the test set segmented with the CC-based technique. On average, it took them 0.30 minutes to post-edit a

[6] http://metav.uab.cat

[7] http://pagines.uab.cat/mtav

[8] http://subworkshop.sourceforge.net/

[9] https://toggl.com/

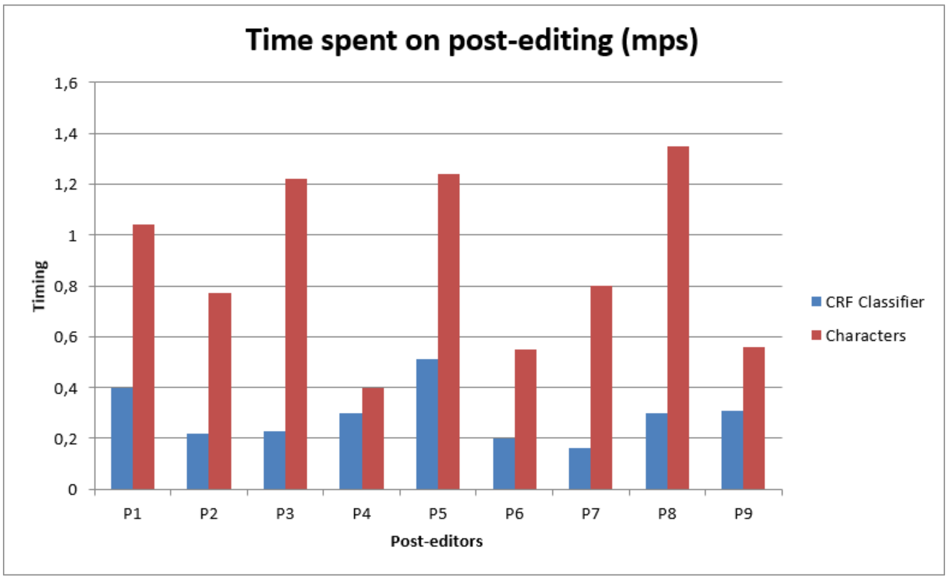

Fig. 2: Productivity evaluation results

subtitle segmented with the CRF-based method and 0.88 minutes to post-edit a subtitle segmented with the counting characters method, which is almost 3 times longer overall. These differences are more noticeable in some cases. For instance, P8 needed, on average, only 0.30 minutes to post-edit a subtitle segmented with the CRF classifier and 1.35 minutes to post-edit a subtitle segmented with the other method under evaluation. It is worth mentioning that P8 was one of the most experimented participants in the manual generation of subtitles.

The results demonstrated that it was much faster to post-edit the subtitles segmented with the CRF classifier, making the post-editing task an easier and more pleasant activity.

9. Conclusions and Future Work

The use of CRF for automatic segmentation of subtitles allowed us to improve the results obtained in the previous work [5] in the following points: (1) differing between the type of breaks (line- and subtitle-breaks), (2) obtaining much better scores and thus generating more and better segmented subtitles and (3) faster processing time. The first point is given by the methodology we employed to construct the CRF models, which was focused on modeling transitions between labels corresponding to each word and its function within the subtitle. The second point was demonstrated in Section 7, in which we showed how the CRF models outperformed the results obtained by the SVM-, Chink-Chunk- and CC-based methods for different types of corpora in Basque and Spanish. For the third point, we presented computation times at subtitle level for the CRF- and SVM-based classification methods, making clear that CRF model (less than 0.1 milliseconds per subtitle) needed much less decoding time than the SVM classification method (16 seconds per subtitle) on similar computers. Finally, a productivity study was presented with human evaluators, which allowed us to show that post-editing subtitles created through the CRF model took less time than to generate them from those obtained by using a more naive method.

The future work will involve experimentation with Recurrent Neural Networks (RNNs) for the task of automatic segmentation. RNNs have been proven to be useful for sequence labeling due to their several and attractive properties, including that they are able to make use of the past and future contextual information, and that they are robust to possible local distorsions of the input sequence [39]. In addition, we will evaluate the performance of the recently generated CRF models with contents of different domains, and more extensively experiments will be performed on bigger datasets, different languages and over the speech recognition output, which may contain unexpected recognition errors. Furthermore, more parameters should be explored to test their impact in this labeling task, such as stop words, syntactic functions or grammatical relations of the different clauses within a sentence. Finally, given that automatic subtitling is an alternative for live broadcasts, for which traditional manual subtitling is less effective, a solution for real-time automatic segmentation should be developed. Considering their computing and decoding time, CRF graphical models would be an interesting solution, but features with low impact in performance, like POS information, should be removed because of the time needed for their computation. Hence, new CRF models should be built including new combinations of different feature sets for the live broadcast environment.

10. Acknowledgements

This work was partially supported by the project CoMUN-HaT - TIN2015-70924-C2-1-R (MINECO/FEDER).

References

[1] A. Álvarez, C. Mendes, M. Raffaelli, T. Luís, S. Paulo, N. Piccinini, H. Arzelus, J. Neto, C. Aliprandi, A. del Pozo, Automating live and batch subtitling of multimedia contents for several european languages, Multimedia Tools and Applications (2015) 1–31.

[2] D. J. Rajendran, A. T. Duchowski, P. Orero, J. Martínez, P. Romero-Fresco, Effects of Text Chunking on Subtitling: A Quantitative and Qualitative Examination, Perspectives 21 (1) (2013) 5–21.

[3] G. D'Ydewalle, J. V. Rensbergen, 13 Developmental Studies of Text-Picture Interactions in the Perception of Animated Cartoons with Text, Advances in Psychology 58 (1989) 233–248.

[4] E. Perego, F. Del Missier, M. Porta, M. Mosconi, The Cognitive Effectiveness of Subtitle Processing, Media Psychology 13 (3) (2010) 243–272.

[5] A. Álvarez, H. Arzelus, T. Etchegoyhen, Towards customized automatic segmentation of subtitles, in: Advances in Speech and Language Technologies for Iberian Languages, Vol. 8854 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 229–238.

[6] V. Warnke, R. Kompe, H. Niemann, E. Nöth, Integrated dialog act segmentation and classification using prosodic features and language models, in: Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997, 1997, pp. 207–210.

[7] I. Read, S. Cox, Stochastic and syntactic techniques for predicting phrase breaks., Computer Speech & Language 21 (3) (2007) 519–542.

[8] D. Beeferman, A. L. Berger, J. D. Lafferty, Cyberpunc: a lightweight punctuation annotation system for speech, in: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98, Seattle, Washington, USA, May 12-15, 1998, 1998, pp. 689–692. doi:10.1109/ICASSP.1998.675358.

[9] J. Mrozinski, E. W. D. Whittaker, P. Chatain, S. Furui, Automatic sentence segmentation of speech for automatic summarization, in: 2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006, 2006, pp. 981–984. doi:10.1109/ICASSP.2006.1660187.

[10] E. Matusov, A. Mauser, H. Ney, Automatic sentence segmentation and punctuation prediction for spoken language translation, in: International Workshop on Spoken Language Translation, Kyoto, Japan, 2006, pp. 158–165.

[11] Y. Gotoh, S. Renals, Sentence boundary detection in broadcast speech transcripts, in: Proc. of ISCA Workshop: Automatic Speech Recognition: Challenges for the new Millennium ASR-2000, 2000, pp. 228–235.

[12] E. Shriberg, A. Stolcke, D. Z. Hakkani-Tür, G. Tür, Prosody-based automatic segmentation of speech into sentences and topics., Speech Communication 32 (1-2) (2000) 127–154.

[13] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, B. Peskin, M. Harper, The icsi-sri-uw metadata extraction system, in: in Proc. of the Intl. Conference on Spoken Language Processing, 2004, pp. 577–580.

[14] Ü. Güz, B. Favre, D. Z. Hakkani-Tür, G. Tür, Generative and discriminative methods using morphological information for sentence segmentation of turkish, IEEE Trans. Audio, Speech & Language Processing 17 (5) (2009) 895–903. doi:10.1109/TASL.2009.2016393.

[15] B. Roark, Y. Liu, M. P. Harper, R. Stewart, M. Lease, M. G. Snover, I. Shafran, B. J. Dorr, J. Hale, A. Krasnyanskaya, L. Yung, Reranking for sentence boundary detection in conversational speech., in: ICASSP (1), IEEE, 2006, pp. 545–548.

[16] T. Oba, T. Hori, A. Nakamura, Sentence boundary detection using sequential dependency analysis combined with crf-based chunking, in: INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006, pp. 1153–1156.

[17] Y. Liu, A. Stolcke, E. Shriberg, M. Harper, Using conditional random fields for sentence boundary detection in speech, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2005, pp. 451–458. doi:10.3115/1219840.1219896.

[18] T. Kawahara, M. Saikou, K. Takanashi, Automatic detection of sentence and clause units using local syntactic dependency, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007, pp. 125–128. doi:10.1109/ICASSP.2007.367179.

[19] F. Gallwitz, H. Niemann, E. Nöth, V. Warnke, Integrated recognition of words and prosodic phrase boundaries., Speech Communication 36 (1-2) (2002) 81–95.

[20] F. Batista, H. Moniz, I. Trancoso, H. Meinedo, A. I. Mata, N. J. Mamede, Extending the punctuation module for european portuguese, in: INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010, pp. 1509–1512.

[21] E. Perego, Subtitles and line-breaks: Towards improved readability, Vol. 78, John Benjamins Publishing, 2008, pp. 211–223.

[22] A. Álvarez, A. Matamala, A. d. Pozo, M. Balenciaga, C. D. Martínez Hinarejos, H. Arzelus, Impact of automatic segmentation on the quality, productivity and self-reported post-editing effort of intralingual subtitles, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), 2016, pp. 3049–3053.

[23] S. Nowozin, C. H. Lampert, Structured learning and prediction in computer vision, Found. Trends. Comput. Graph. Vis. 6 (3-4) (2011) 185–365. doi:10.1561/0600000033.
URL `http://dx.doi.org/10.1561/0600000033`

[24] Y. Liu, J. Carbonell, P. Weigele, V. Gopalakrishnan, Protein fold recognition using segmentation conditional random fields (scrfs), Journal of Computational Biology 13 (2) (2006) 394–406.

[25] A. Gunawardana, M. Mahajan, A. Acero, J. C. Platt, Hidden conditional random fields for phone classification, in: Interspeech, 2005, pp. 1117–1120.

[26] A. McCallum, W. Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, in: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, CONLL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, 2003, pp. 188–191. doi:10.3115/1119176.1119206.
URL `http://dx.doi.org/10.3115/1119176.1119206`

[27] D. Roth, W.-t. Yih, Integer linear programming inference for conditional random fields, in: Proceedings of the 22nd International Conference on Machine Learning, ICML '05, ACM, New York, NY, USA, 2005, pp. 736–743. doi:10.1145/1102351.1102444.
URL `http://doi.acm.org/10.1145/1102351.1102444`

[28] F. Sha, F. Pereira, Shallow parsing with conditional random fields, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, 2003, pp. 134–141. doi:10.3115/1073445.1073473.
URL `http://dx.doi.org/10.3115/1073445.1073473`

[29] F. Peng, F. Feng, A. McCallum, Chinese segmentation and new word detection using conditional random fields, in: Proceedings of the 20th International Conference on Computational Linguistics, COLING '04, Association for Computational Linguistics, Stroudsburg, PA, USA, 2004, pp. 562–568. doi:10.3115/1220355.1220436.
URL `http://dx.doi.org/10.3115/1220355.1220436`

[30] C.-D. Martínez-Hinarejos, J.-M. Benedí, V. Tamarit, Unsegmented dialogue act annotation and decoding with n-gram transducers, IEEE/ACM Transactions on Audio, Speech, and Language Processing 23 (1) (2015) 198–211.

[31] C. Sutton, A. McCallum, An introduction to conditional random fields, Foundations and Trends in Machine Learning 4 (4) (2012) 267–373.

[32] G. B. Flores d'Arcais, Syntactic processing during reading for comprehension., Lawrence Erlbaum Associates, Inc, 1987.

[33] M. E. Coltheart, Attention and performance 12: The psychology of reading., Lawrence Erlbaum Associates, Inc, 1987.

[34] N. Ezeiza, I. Alegria, J. M. Arriola, R. Urizar, I. Aduriz, Combining stochastic and rule-based methods for disambiguation in agglutinative languages, in: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 1998, pp. 380–384.

[35] R. Agerri, J. Bermudez, G. Rigau, Multilingual, Efficient and Easy NLP Processing with IXA Pipeline, in: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, pp. 5–8.

[36] A. Álvarez, P. Ruiz, H. Arzelus, Improving a long audio aligner through phone-relatedness matrices for english, spanish and basque, in: P. Sojka, A. Horák, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue, Vol. 8655 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 473–480.

[37] T. Kudo, Crf++: Yet another crf toolkit, Software available at http://crfpp. sourceforge. net.

[38] NIST, Nist website: Rt-03 fall rich transcription, `http://www.itl.nist.gov/iad/mig/tests/rt/2003-fall/index.html` (2003).

[39] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Vol. 385 of Studies in Computa-

tional Intelligence, Springer, 2012. doi:10.1007/978-3-642-24797-2.
URL `http://dx.doi.org/10.1007/978-3-642-24797-2`