Exploiting the human computational effort dedicated to message reply formatting for training discursive email segmenters

Nicolas Hernandez   **Soufian Salim**
LINA UMR 6241 Laboratory
University of Nantes (France)
firstname.lastname@univ-nantes.fr

Abstract

In the context of multi-domain and multimodal online asynchronous discussion analysis, we propose an innovative strategy for manual annotation of dialog act (DA) segments. The process aims at supporting the analysis of messages in terms of DA. Our objective is to train a sequence labelling system to detect the segment boundaries. The originality of the proposed approach is to avoid manually annotating the training data and instead exploit the human computational efforts dedicated to message reply formatting when the writer replies to a message by inserting his response just after the quoted text appropriate to his intervention. We describe the approach, propose a new electronic mail corpus and report the evaluation of segmentation models we built.

1 Introduction

Automatic processing of online conversations (forum, emails) is a highly important issue for the industrial and the scientific communities which care to improve existing question/answering systems, identify emotions or intentions in customer requests or reviews, detect messages containing requests for action or unsolved severe problems...

In most works, conversation interactions between the participants are modelled in terms of dialogue acts (DA) (Austin, 1962). The DAs describe the communicative function conveyed by each text utterance (e.g. question, answer, greeting,...). In this paper, we address the problem of rhetorically segmenting the new content parts of messages in online asynchronous discussions. The process aims at supporting the analysis of messages in terms of DA. We pay special attention to the processing of electronic mails.

The main trend in automatic DA recognition consists in using supervised learning algorithms to predict the DA conveyed by a sentence or a message (Tavafi et al., 2013). The hypothesized message segmentation results from the global analysis of these individual predictions over each sentence. A first remark on this paradigm is that it is not realistic to use in the context of multi-domain and multimodal processing because it requires the building of training data which is a very substantial and time-consuming task. A second remark is that the model does not have a fine-grained representation of the message structure or the relations between messages. Considering such characteristics could drastically improve the systems to allow to focus on specific text parts or to filter out less relevant ones. Indeed, apart from the closing formula, a message may for example be made of several distinct information requests, the description of an unsuccessful procedure, the quote of third-party messages...

So far, few works address the problem of message segmentation. (Lampert et al., 2009a) propose to segment emails in prototypical zones such as the author's contribution, quotes of original messages, the signature, the opening and closing formulas. In comparison, we focus on the segmentation of the author's contribution (what we call the new content part). (Joty et al., 2013) identifies clusters of topically related sentences through the multiple messages of a thread, without distinguishing email and forum messages. Apart from the topical aspect, our problem differs because we are only interested in the cohesion between sentences in nearby fragments and not on distant sentences.

LAW VIII - The 8th Linguistic Annotation Workshop, pages 110–119,
Dublin, Ireland, August 23-24 2014.

[Hi!]^{S1}

[I got my ubuntu cds today and i'm really impressed.]^{S2} [My
friends like them and my teachers too (i'm a student).]^{S3}
[It's really funny to see, how people like ubuntu and start feeling geek
and blaming microsoft when they use it.]^{S4}

[Unfortunately everyone wants an ubuntu cd, so can i download the cd
covers anywhere or an 'official document' which i can attach to
self-burned cds?]^{S5}

[I searched the entire web site but found nothing.]^{S6} [Thanks in advance.]^{S7}

[John]^{S8}

(a) Original message.

[On Sun, 04 Dec 2005, John Doe <john@doe.com> wrote:]^{R1}

> [I got my ubuntu cds today and i'm really impressed.]^{R2} [My
> friends like them and my teachers too (i'm a student).]^{R3}
> [It's really funny to see, how people like ubuntu and start feeling geek
> and blaming microsoft when they use it.]^{R4}

[Rock!]^{R5}

> [Unfortunately everyone wants an ubuntu cd, so can i download the cd
> covers anywhere or an 'official document' which i can attach to
> self-burned cds?]^{R6}

[We don't have any for the warty release, but we will have them for hoary, because quite a few people have asked. :-)]^{R7}

[Bob.]^{R8}

(b) Reply message.

Figure 1: An original message and its reply (*ubuntu-users* email archive). Sentences have been tagged to facilitate the discussion.

Original	**Reply**	**Label**
S1		
	R1	
S2	> *R2*	`Start`
S3	> *R3*	`Inside`
S4	> *R4*	`End`
	R5	
S5	> *R6*	`Start&End`
	R7	
	[...]	
S6		
[...]		

Figure 2: Alignment of the sentences from the original and reply messages shown in Figure 1 and labels inferred from the re-use of the original message text. Labels are associated to the original sentences.

Despite the drawbacks mentioned above, a supervised approach remains the most efficient and reliable method to solve classification problems in Natural Language Processing. Our aim is to train a system to detect the segment boundaries, i.e. to determine, through a classification approach, if a given sentence starts, ends or continues a segment.

The originality of the proposed approach is to avoid manually annotating the training data and instead to exploit the human computational efforts dedicated to a similar task in a different context of production (von Ahn, 2006). As recommended by the *Netiquette*[1], when replying to a message (email or forum post), the writer should "summarize the original message at the top of its reply, or include (or "quote") just enough text of the original to give a context, in order to make sure readers understand when they start to read the response[2]." As a corollary, the writer should "edit out all the irrelevant material." Our idea is to use this effort, in particular when the writer replies to a message by inserting his response or comment just after the quoted text appropriate to his intervention. This posting style is called *interleaved* or *inline replying*. The so built segmentation model should be usable for any posting styles by applying it only on new content parts. Figure 1a shows an example of an *original* message and, Figure 1b, one of its *reply*. We can see that the reply message re-uses only four selected sentences from the original message; namely $S2$, $S3$, $S4$ and $S5$ which respectively correspond to sentences $R2$, $R3$, $R4$ and $R6$ in the reply message. The author of the reply message deliberately discarded the remaining of the original message. The segment build up by sentences $S2$, $S3$, $S4$ and the one by the single sentence $S5$ can respectively be associated with two acts : a comment and a question.

In Section 2, we explain our approach for building an annotated corpus of segmented online messages at no cost. In Section 3, we describe the system and the features we use to model the segmentation. After

[1] Set of guidelines for Network Etiquette (*Netiquette*) when using network communication or information services RFC1855.

[2] It is true that some email software clients do not conform to the recommendations of Netiquette and that some online participants are less sensitive to arguments about posting style (many writers reply above the original message). We assume that there are enough messages with inline replying available to build our training data.

presenting our experimental framework in Section 4, we report some evaluations for the segmentation task in Section 5. Finally, we discuss our approach in comparison to other works in Section 6.

2 Building annotated corpora of segmented online discussions at no cost

We present the assumptions and the detailed steps of our approach.

2.1 Annotation scheme

The basic idea is to interpret the operation performed by a discussion participant on the message he replies as an annotation operation. Assumptions about the kind of annotations depend on the operation that has been performed. Deletion or re-use of the original text material can give hints about the relevance of the content: discarded material is probably less relevant than re-used one.

We assume that by replying inside a message and by only including some specific parts, the participant performs some cognitive operations to identify homogeneous self-contained text segments. Consequently, we make some assumptions about the role played by the sentences in the original message information structure. A sentence in a segment plays one of the following roles: `starting and ending` (*SE*) a segment when there is only one sentence in the segment, `starting` (*S*) a segment if there are at least two sentences in the segment and it is the first one, `ending` (*E*) a segment if there are at least two sentences in the segment and it is the last one, `inside` (*I*) a segment in any other cases.

Figure 2 illustrates the scheme by showing how sentences from Figure 1 can be aligned and the labels inferred from it. It is similar to the *BIO* scheme except it is not at the token level but at the sentence level (Ratinov and Roth, 2009).

2.2 Annotation generation procedure

Before being able to predict labels of the original message sentences, it is necessary to identify those that are re-used in a reply message. Identification of the quoted lines in a reply message is not sufficient for various reasons. First, the segmenter is intended to work on non-noisy data (i.e. the new content parts in the messages) while a quoted message is an altered version of the original one. Indeed, some email software clients involved in the discussion are not always standards-compliant and totally compatible[3]. In particular, the quoted parts can be wrongly re-encoded at each exchange step due to the absence of dedicated header information. In addition, the client programs can integrate their own mechanisms for quoting the previous messages when including them as well as for wrapping too long lines[4]. Second, accessing the original message may allow taking some contextual features into consideration (like the visual layout for example). Third, to go further, the original context of the extracted text also conveys some segmentation information. For instance, a sentence from the original message, not present in the reply, but following an aligned sentence, can be considered as starting a segment.

So in addition to identifying the quoted lines, we deploy an alignment procedure to get the original version of the quoted text. In this paper, we do not consider the contextual features from the original message and focus only on sentences that have been aligned.

The generation procedure is intended to "automatically" annotate sentences from the original messages with segmentation information. The procedure follows the following steps:

1. Messages posted in the interleaved replying style are identified
2. For each pair of original and reply messages:
   (a) Both messages are tokenized at sentence and at word levels
   (b) Quoted lines in the reply message are identified
   (c) Sentences which are part of the quoted text in the reply message are identified

[3] The *Request for Comments* (RFC) are guidelines and protocols proposed by working groups involved in the Internet Standardization `https://tools.ietf.org/html`, the message contents suffer from encoding and decoding problems. Some of the RFC are dedicated to email format and encoding specifications (See RFC 2822 and 5335 as starting points). There have been several propositions with updates and consequently obsoleted versions which may explain some alteration issues.

[4] Feature for making the text readable without any horizontal scrolling by splitting lines into pieces of about 80 characters.

(d) Sentences in the original message are aligned with quoted text in the reply message [5]

(e) Aligned original sentences are labelled in terms of position in segment

(f) The sequence of labelled sentences is added to the training data

Messages with *inline replying* are recognized thanks to the presence of at least two consecutive quoted lines separated by new content lines. Pairs of original and reply messages are constituted based on the `in-reply-to` field present in the email headers. As declared in the RFC 3676[6], we consider as *quoted lines*, the lines beginning with the ">" (greater than) sign. Lines which are not quoted lines are considered to be *new content* lines. The word tokens are used to index the quoted lines and the sentences.

Labelling of aligned sentence (sentence from the original message re-used in the reply message) uses this simple rule-based algorithm:

```
For each aligned original sentence:
    if the sentence is surrounded by new content in the reply message, the label is Start&End
    else if the sentence is preceded by a new content, the label is Start
    else if the sentence is followed by a new content, the label is End
    else, the label is Inside
```

2.3 Alignment module

For finding alignments between two given text messages, we use a *dynamic programming (DP) string alignment algorithm* (Sankoff and Kruskal, 1983). In the context of speech recognition, the algorithm is also known as the *NIST align/scoring algorithm*. Indeed, it is widely used to evaluate the output of speech recognition systems by comparing the hypothesized text output by the speech recognizer to the correct, or reference text. The algorithm works by "performing a global minimization of a Levenshtein distance function which weights the cost of correct words, insertions, deletions and substitutions as 0, 75, 75 and 100 respectively. The computational complexity of DP is $O(MN)$."

The Carnegie Mellon University provides an implementation of the algorithm in its speech recognition toolkit[7]. We use an adaptation of it which allows working on lists of strings[8] rather than directly on strings (as sequences of characters).

3 Building the segmenter

Each email is processed as a sequence of sentences. We choose to define the segmentation problem as a sequence labelling task whose aim is to assign the globally best set of labels for the entire sequence at once. The underlying idea is that the choice of the optimal label for a given sentence is dependent on the choices of nearby sentences. Our email segmenter is built around a linear-chain Conditional Random Field (CRF), as implemented in the sequence labelling toolkit Wapiti (Lavergne et al., 2010).

Training the classifier to recognize the different labels of the previously defined annotation scheme can be problematic. It has indeed some disadvantages that can undermine the effectiveness of the classifier. In particular, sentences annotated *SE* will, by definition, share important characteristics with sentences bearing the annotation *S* and *E*. So we chose to transform these annotations into a binary scheme and merely differentiate sentences that starts a new segment (*True*), or "boundary sentences", from those that do not (*False*). The conversion process is trivial, and can easily be reversed[9].

We distinguish four sets of features: n-gram features, information structure based features, thematic features and miscellaneous features. All the features are domain-independent. Almost all features are language-independent as well, save for a few that can be easily translated. For our experiments, the CRF window size is set at 5, i.e. the classification algorithm takes into account features of the next and previous two sentences as well as the current one.

[5] Section 2.3 details how alignment is performed.

[6] `http://www.ietf.org/rfc/rfc3676.txt`

[7] Sphinx 4 `edu.cmu.sphinx.util.NISTAlign` `http://cmusphinx.sourceforge.net`

[8] `https://github.com/romanows/WordSequenceAligner`

[9] Sentences labelled with *SE* or *S* are turned into *True*, the other ones into *False*. To reverse the process, a *True* is turned into *SE* if the next sentence is also a boundary (i.e. a True) and into *S* otherwise. While a *False* is turned into *E* if the next sentence is a boundary (i.e. a True) and into *I* otherwise.

n-gram features We select the case-insensitive word bi-grams and tri-grams with the highest document frequency in the training data (empirically we select the top 1,000 n-grams), and check for their presence in each sentence. Since the probability of having multiple occurrences of the same n-gram in one sentence are extremely low, we do not record the number of occurrences but merely a boolean value.

Information structure based features This feature set is inspired by the information structure theory (Kruijff-Korbayová and Kruijff, 1996) which describes the information imparted by the sentence in terms of the way it is related to prior context. The theory relates these functions with particular syntactic constructions (e.g. topicalization) and word order constraints in the sentence.

We focus on the first and last three *significant* tokens in the sentence. A token is considered as significant if its occurrence frequency is higher than 1/2,000[10]. As features we use n-grams of the surface form, lemma and part-of-speech tag of each triplet (36 features).

Thematic feature The only feature we use to account for thematic shift recognition is the output of the TextTiling algorithm (Hearst, 1997). TextTiling is one of the most commonly used algorithms for automatic text segmentation. If the algorithm detects a rupture in the lexical cohesion of the text (between two consecutive blocks), it will place a boundary to indicate a thematic change. Due to the short size of the messages, we define a block size to equate the sum of three times the sentence average size in our corpus. We set the step-size (overlap size of the rolling window) to the average size of a sentence.

Miscellaneous features This feature set includes stylistic and semantic features. 24 features, several of them borrowed from related work in speech act classification (Qadir and Riloff, 2011) and email segmentation (Lampert et al., 2009b), are in the set: *Stylistic features* capture information about the visual structure and composition of the message: the position of the sentence in the email, the average length of a token, the total number of tokens and characters, the proportion of upper-case, alphabetic and numeric characters, the number of greater-than signs (">"); whether the sentence ends with or contains a question mark, a colon or a semicolon; whether the sentence contains any punctuation within the first three tokens (this is meant to recognize greetings (Qadir and Riloff, 2011)).

Semantic features check for meaningful words and phrases: whether the sentence begins with or contains a "wh*" question word or a phrase suggesting an incoming interrogation (e.g. *"is it"*, *"are there"*); whether the sentence contains a modal; whether any plan phrases (e.g. *"i will"*, *"we are going to"*) are present; whether the sentence contains first person (e.g. *"we"*, *"my"*) second person or third person words; the first personal pronoun found in the sentence; the first verbal form found.

4 Experimental framework

We describe the data, the preprocessing and the evaluation protocol we use for our experiments.

4.1 Corpus

The current work takes place in a project dealing with multilingual and multimodal discussion processing, mainly in interrogative technical domains. For these reasons we did not consider the Enron Corpus (30,000 threads) (Klimt and Yang, 2004) (which is from a corporate environment), neither the W3C Corpus (despite its technical consistence) or its subset, the British Columbia Conversation Corpus (BC3) (Ulrich et al., 2008).

We rather use the *ubuntu-users* email archive[11] as our primary corpus. It offers a number of advantages. It is free, and distributed under an unrestrictive license. It increases continuously, and therefore is representative of modern emailing in both content and formatting. Additionally, many alternatives archives are available, in a number of different languages, including some very resource-poor languages. Ubuntu also offers a forum and a FAQ which are interesting in the context of multimodal studies.

We use a copy of December 2013. The corpus contains a total of 272,380 messages (47,044 threads). 33,915 of them are posted in the inline replying style that we are interested in. These messages are made

[10] This value was set up empirically on our data. More experimentation needs to be done to generalize it.

[11] Ubuntu mailing lists archives (See *ubuntu-users*): `https://lists.ubuntu.com/archives/`

of 418,858 sentences, themselves constituted of 76,326 unique tokens (5,139,123 total). 87,950 of these lines (21%) are automatically labelled by our system as the start of a new segment (either *SE* or *S*).

4.2 Evaluation protocol

In order to evaluate the efficiency of the segmenter, we perform a 10-fold cross-validation on the Ubuntu corpus, and compare its performance to two different baselines. The first one, the "regular" baseline, is computed by segmenting the test set into regular segments of the same length as the average training set segment length, rounded up. The second one is the TextTiling algorithm we described in section 3. While it is used as a feature in the proposed approach in the previous section, the direct output of the TextTiling algorithm is used for the baseline.

The results are measured with a panel of metrics used in text segmentation and Information Retrieval (IR). Precision (P) and Recall (R) are provided for all results. P is the percentage of boundaries identified by the classifier that are indeed true boundaries. R is the percentage of true boundaries that are identified by the classifier. We also provide the harmonic mean of precision and recall: $F_1 = 2 \cdot \frac{P \cdot R}{P+R}$

However, automatic evaluation of speech segmentation through these metrics is problematic as predicted segment boundaries seldom align precisely. Therefore, we also provide an array of metrics relevant to the field of text segmentation : P_k, *WindowDiff* and the *Generalized Hamming Distance (GHD)*. The P_k metric is a probabilistically motivated error metric for the assessment of segmentation algorithms (Beeferman et al., 1999). *WindowDiff* compares the number of segment boundaries found within a fixed-sized window to the number of boundaries found in the same window of text for the reference segmentation (Pevzner and Hearst, 2002). The *GHD* is an extension of the Hamming distance[12] that gives partial credit for near misses (Bookstein et al., 2002).

4.3 Preprocessing

To reduce noise in the corpus we filter out undesirable emails based on several criteria, the first of which is encoding. Messages that are not UTF-8 encoded are removed from the selection. The second criterion is MIME type: we keep single-part plain text messages only, and remove those with HTML or other special contents. In addition, we choose to consider only replies to thread starters. This choice is based on the assumption that the alignment module would have more difficulty in recognizing properly sentences that were repeatedly transformed in successive replies. Indeed, these replies - that would contain quoted text from other messages - would be more likely to be poorly labelled through automatic annotation. The last criterion is length. The dataset being built from a mailing list that can cover very technical discussions, users sometimes send very lengthy messages containing many lines of copied-and-pasted code, software logs, bash command outputs, etc. The number of these messages is marginal, but their lengths being disproportionately high, they can have a negative impact on the segmenter's performance. We therefore exclude messages longer than the average message length plus the standard length deviation. After filtering, the dataset is left with 6,821 messages out of 33,915 (20%).

For building the segmenter features, we use the Stanford Part-Of-Speech Tagger for morpho-syntactic tagging (Toutanova et al., 2003), and the WordNet lexical database for lemmatization (Miller, 1995).

5 Experiments

Table 1 shows the summary of all obtained results. On the left side are shown results about segmentation metrics, on the right side results about information retrieval metrics. First, we examine baseline scores, and display them in the top section. Second, in the middle section, we show results for segmenters based on individual feature sets (with A standing for n-grams, B for information structure, C for TextTiling and D for miscellaneous features). Finally, in the lower section, we show results based on feature sets combinations.

[12] Wikipedia article on the Hamming distance: `http://en.wikipedia.org/wiki/Hamming_distance`

	Segmentation metrics			Information Retrieval metrics		
	WD	P_k	GHD	P	R	F_1
regular baseline	.59	.25	.60	.31	.49	.38
TextTiling baseline	.41	.07	.38	.75	.44	.56
$\phi(A)$ with A = n-grams	.38	**.05**	.39	**1**	.39	.56
$\phi(B)$ with B = info. structure	.43	.11	.38	.60	.68	**.64**
$\phi(C)$ with C = TextTiling	.39	.05	.38	.94	.40	.56
$\phi(D)$ with D = misc. features	.41	.09	.38	.69	.49	.57
$\phi(A+B+C+D)$	.38	**.05**	.39	**1**	.39	.56
$\phi(\phi(A)+\phi(B)+\phi(C)+\phi(D))$	.38	.06	.36	.81	.47	.59
$\phi(A) \cup \phi(B+C+D)$	.45	.12	.40	.58	**.69**	.63
$\phi(A) \cup \delta(\phi(B+C+D))$	**.36**	.06	**.34**	.80	.53	**.64**

Table 1: Comparative results between baselines and tested segmenters. All displayed results show *WindowDiff* (*WD*), P_k and *GHD* as error rates, therefore a lower score is desirable for these metrics. This contrasts with the three IR scores, for which a low value denotes poor performance. Best scores are shown in bold.

5.1 Baseline segmenters

The first section of Table 1 shows the results obtained by both of our baselines. Unsurprisingly, TextTiling performs much better than the basic regular segmentation algorithm across all metrics save recall.

5.2 Segmenters based on individual feature sets

The second section of Table 1 shows the results for four different classifiers, each trained with a distinct subset of the feature set. The ϕ function is the classification function, its parameters are features, and its output a prediction. While all classifiers easily beat the regular baseline, and match the TextTiling baseline when it comes to IR metrics, only the thematic and the n-grams segmenters manage to surpass TextTiling when performance is measured by segmentation metrics. In terms of IR scores, the n-grams classifier in particular stands out as it manages to achieve an outstanding 100% precision, although this result is mitigated by a meager 39% recall. It is also interesting to see that the thematic classifier, based only on contextual information about TextTiling output, performs better than the TextTiling baseline.

5.3 Segmenters based on feature sets combinations

The last section of Table 1 shows the results of four different segmenters. The first one, $\phi(A{+}B{+}C{+}D)$, is a simple classifier that takes all available features into account. Its results are exactly identical to that of the n-grams classifier, most certainly due to the fact that other features are filtered out due to the sheer number of lexical features. The second one, $\phi(\phi(A) + \phi(B) + \phi(C) + \phi(D))$, uses as features the outputs of the four classifiers trained on each individual feature set. Results show this approach isn't significantly better. The third one, $\phi(A) \cup \phi(B + C + D)$, segments according to the union of the boundaries detected by a classifier trained on n-grams features and those identified by a classifier trained on all other features. This idea is motivated by the fact that we know all boundaries found by the n-grams classifier to be accurate ($P = 1$). Doing this allows the segmenter to obtain the best possible recall ($R = .69$), but at the expense of precision ($P = .58$). The last one, $\phi(A) \cup \delta(\phi(B + C + D))$, attempts to increase the n-grams classifier's recall without sacrificing too much precision by being more selective about boundaries. The δ function is the "cherry picking" function, which filters out boundaries predicted without sufficient confidence. Only those identified by the n-grams classifier and those classified as boundaries with a confidence score of at least .99 by a classifier trained on the other feature sets are considered. This system outperforms all others both in terms of segmentation scores and F_1, however it is still relatively conservative and the segmentation ratio (the number of guessed boundaries divided by the number of true boundaries) remains significantly lower than expected, at 0.67. Tuning the minimum confidence score (c) allows to adjust P from .58 ($c = 0$) to 1 ($c = 1$) and R from .39 ($c = 1$) to .69 ($c = 0$).

6 Related work

Three research areas are directly related to our study: a) collaborative approaches for acquiring annotated corpora, b) detection of email structure, and c) sentence alignment. In the (Wang et al., 2013)'s taxonomy of the collaborative approaches for acquiring annotated corpora, our approach could be related to the *Wisdom of the Crowds* (WotC) genre where motivators are altruism or prestige to collaborate for the building of a public resource. As a major difference, we did not initiate the annotation process and consequently we did not define annotation guidelines, design tasks or develop tools for annotating which are always problematic questions. We have just rerouted *a posteriori* the result of an existing task which was performed in a distinct context. In our case the burning issue is to determine the adequacy of our segmentation task. Our work is motivated by the need to identify important snippets of information in messages for applications such as being able to determine whether all the aspects of a customer request were fully considered. We argue that even if it is not always obvious to tag topically or rhetorically a segment, the fact that it was a human who actually segmented the message ensures its quality. We think that our approach can also be used for determining the relevance of the segments, however it has some limits, and we do not know how labelling segments with dialogue acts may help us do so.

Detecting the structure of a thread is a hot topic. As mentioned in Section 1, very little works have been done on email segmentation. We are aware of recent works in linear text segmentation such as (Kazantseva and Szpakowicz, 2011) who addresses the problem by modelling the text as a graph of sentences and by performing clustering and/or cut methods. Due to the size of the messages (and consequently the available lexical material), it is not always possible to exploit this kind of method. However, our results tend to indicate that we should investigate in this direction nonetheless. By detecting sub-units of information within the message, our work may complement the works of (Wang et al., 2011; Kim et al., 2010) who propose solutions for detecting links between messages. We may extend these approaches by considering the possibility of pointing from/to multiple message sources/targets.

Concerning the alignment process, our task can be compared to the detection of monolingual text derivation (otherwise called plagiarism, near–duplication, revision). (Poulard et al., 2011) compare, for instance, the use of n–grams overlap with the use of text hapax. In contrast, we already know that a text (the reply message) derives from another (the original message). Sentence alignment has also been a very active field of research in statistical machine translation for building parallel corpora. Some methods are based on sentence length comparison (Gale and Church, 1991), some methods rely on the overlap of rare words (cognates and named entities) (Enright and Kondrak, 2007). In comparison, in our task, despite some noise, the compared text includes large parts of material identical to the original text. The kinds of edit operation in presence (no inversion[13] only deletion, insertion and substitution) lead us to consider the Levenshtein distance as a serious option.

7 Future work

The main contribution of this work is to exploit the human effort dedicated to reply formatting for training discursive email segmenters. We have implemented and tested various segmenter models. There is still room for improvement, but our results indicate that the approach merits more thorough examination. Our segmentation approach remains relatively simple and can be easily extended. One way would be to consider contextual features in order to characterize the sentences in the original message structure. As future works, we plan to complete our current experiments with two new approaches for evaluation. The first one will consists in comparing the automatic segmentation with those performed by human annotators. This task remains tedious since it will then be necessary to define an annotation protocol, write guidelines and build other resources. The second evaluation we plan to perform is an extrinsic evaluation. The idea will be to measure the contribution of the segmentation in the process of detecting the dialogue acts, i.e. to check if existing sentence-level classification systems would perform better with such contextual information.

[13] When computing the Levenshtein distance, the inversion edit operation is the most costly operation.

References

John L. Austin. 1962. *How to do Things with Words: The William James Lectures delivered at Harvard University in 1955.* Oxford: Clarendon Press.

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

Abraham Bookstein, Vladimir A Kulyukin, and Timo Raita. 2002. Generalized hamming distance. *Information Retrieval*, 5(4):353–375.

Jessica Enright and Grzegorz Kondrak. 2007. A fast method for parallel document identification. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 29–32, Rochester, New York, April. Association for Computational Linguistics.

William A. Gale and Kenneth Ward Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting of the Association of Computational Linguistics (ACL).*

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

Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2013. Topic segmentation and labeling in asynchronous conversations. *Journal of AI Research (JAIR)*, 47:521–573.

Anna Kazantseva and Stan Szpakowicz. 2011. Linear text segmentation using affinity propagation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 284–293, Stroudsburg, PA, USA. Association for Computational Linguistics.

Su Nam Kim, Li Wang, and Timothy Baldwin. 2010. Tagging and linking web forum posts. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 192–202, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *ECML*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226. Springer.

Ivana Kruijff-Korbayová and Geert-Jan M. Kruijff. 1996. Identification of topic-focus chains. In S. Botley, J. Glass, T. McEnery, and A. Wilson, editors, *Approaches to Discourse Anaphora: Proceedings of the Discourse Anaphora and Anaphora Resolution Colloquium (DAARC96)*, volume 8, pages 165–179. University Centre for Computer Corpus Research on Language, University of Lancaster, UK, July 17-18.

Andrew Lampert, Robert Dale, and Cécile Paris. 2009a. Segmenting email message text into zones. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 919–928, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrew Lampert, Robert Dale, and Cécile Paris. 2009b. Segmenting email message text into zones. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 919–928. Association for Computational Linguistics.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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

Fabien Poulard, Nicolas Hernandez, and Béatrice Daille. 2011. Detecting derivatives using specific and invariant descriptors. *Polibits*, (43):7–13.

Ashequl Qadir and Ellen Riloff. 2011. Classifying sentences as speech acts in message board posts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 748–758. Association for Computational Linguistics.

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 6.

D Sankoff and J B Kruskal. 1983. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts. ISBN 0-201-07809-0.

Maryam Tavafi, Yashar Mehdad, Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2013. Dialogue act recognition in synchronous and asynchronous conversations. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2013)*, SIGDIAL'13.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

J. Ulrich, G. Murray, and G. Carenini. 2008. A publicly available annotated corpus for supervised email summarization. In *AAAI08 EMAIL Workshop*, Chicago, USA. AAAI.

L. von Ahn. 2006. Games with a purpose. *Computer*, 39(6):92–94.

Li Wang, Diana Mccarthy, and Timothy Baldwin. 2011. Predicting thread linking structure by lexical chaining. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 76–85, Canberra, Australia, December.

Aobo Wang, CongDuyVu Hoang, and Min-Yen Kan. 2013. Perspectives on crowdsourcing annotations for natural language processing. *Language Resources and Evaluation*, 47(1):9–31.