Text Segmentation Using Light Syntax Parsing and Fuzzy Systems

Omar Ali[1], Alexander Gegov[1], Ella Haig[1], and Rinat Khusainov[1]

University of Portsmouth, Portsmouth Hampshire PO1 3HE, UK
{omar.ali1,alexander.gegov,ella.haig,rinat.khusainov}@port.ac.uk

Abstract. We present our positions and proposition in regards to the process of text segmentation. The method discussed below aims to improve upon the current processes available for text segmentation by introducing the concept of fuzzy boundaries. our method of segmenting text concerns a population of boundaries to be in a fuzzy state whereby the decision of boundary insertion is determined by a fuzzy system with syntactic information serving as the inputs. Furthermore, we aim to build on this method in the future with the goal of presenting a multifaceted segmentation approach that is applicable across various domains that require segmentation: Text summarisation, Rhetorical structure theory, sentence-based segmentation and paragraph-based segmentation. The following work is presented as follows: we first outline the previously established concepts that were used to develop our understanding of segmentation and fuzzy system. We then move to describe the motivations and our overall position and reasons for the proposed method. The model its components and the validation metric are then outlined with descriptions on how the inputs for the fuzzy system, together with the fuzzy system's architecture, described in detail. We conclude with applications of the segmentation model together with the further work to be carried out on to of this model.

Keywords: Fuzzy Systems · Text Segmentation · Text pre-processing

1 Introduction

1.1 Linear Text Segmentation

Linear text segmentation describes the process of partitioning a text block into sections, or segments[1], such that each segment is coherent. The problem can be decomposed into two types, topical segmentation, and EDU segmentation. Topical segmentation's uses changes in topic throughout the text to inform the segmentation process. If changes in topic are drastic between two adjacent text spans, a boundary is inserted. EDU segmentation follows the rules defined first and foremost in [10]. Within topical segmentation's, the probabilities of segmentation can be seen as the optimisation of two particular parameters: Internal

[1] We will use segment and sentence interchangeably in this work as theoretically, a sentence can be treated exactly the same as a segment

Coherence; the level of coherence within the segment, and External Dissimilarity; the level of dissimilarity between adjacent segments. If these parameters are maximised, we can consider the segment in question to be *valid*. Application of linear text segmentation typically revolve around text summarisation. Text summarisation is the process of reducing an input-text down to the most relevant parts, removing as much *noisy-text* as possible. Another application that has seen the use of linear text segmentation is Rhetorical Structure Theory (RST). The authors of [7] envision the problem differently to the above-described definition. They see the problem as a sequential labelling problem whereby the solution is obtained through a sequence of yes/no tags at the sentence-level. We adopt a similar perspective to the authors in our proposed method; inspired, partially, also by text tiling.

1.2 Methods of Processing Text

Text-Tiling Text-tiling [5] is an algorithm in which we can consider all possible boundaries between every known segment, typically sentence boundaries, and compare a window of words or sentences on either side to evaluate whether the currently considered boundary is a true boundary. Given the scope and desired direction for this work, rhetorical structure theory, we intend to look at boundaries beyond the sentence-level to account for inter-sentence or cause boundaries.

Fuzzy Systems Fuzzy systems describe a model which is capable of transposing typically subjective interpretations of a given concept into crisps outputs. Firstly, one must define a set of membership functions for each crisp input. Each membership function pertains to a feature of the given concept. For example, if the concept in question deals with temperature, the features of this concept could be *hot* or *cold*. We must define the membership functions in such a way that approximates the characteristic of the feature in the concept; for example, *hot* can be defined as as any temperature from 40 to 100 degrees or further with *cold* being defined as any temperature from 40 down to 0 or further. These membership functions can take the form multiple different shapes i.e. trapezoidal, triangular, Gaussian etc.

Phrase-Structure Parsing Phrase-structure parsing, or syntax parsing (Used interchangeably in this work) describes the process of abstract input text into a binary-tree structure. The leaves of the syntax parse tress therefore denote the words in the sentence (in order left to right) and the inner nodes determine the syntax label connecting subtrees together.

1.3 Preliminary Survey of Previous Work

Syntactic information provides us with key information as to the structure of text. Provided we can illicit POS of words in a given sentence, we can generate the syntax tree for that sentence. Authors of [8] and [6] have employed the use

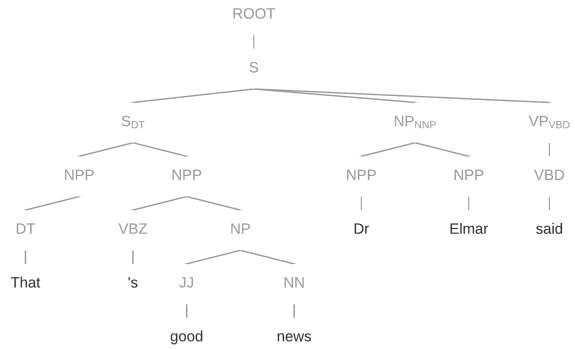

Figure. 1. Parse Tree Complex Example

of syntactic patterns to infer coherence in text. The former authors define a set of syntactic patterns that correspond to a level of cohesion within the text. The latter author, however, uses syntactic cues to prompt segmentation together with cue-phrasing. Whilst this method can be particularly manual, we can see similar methods established by the authors of [14]. [15] uses clause segmentation to improve text summarisation. They use 2 heuristics, one that contains regexes to prompt a possible segmentation, and the second one to evaluate the applicability of the segment proposed i.e. segment length, etc. Authors finally express a need for automated methods including machine learning approaches. In regard to the use of fuzzy systems for text processing, little has been done in terms of the use of fuzzy systems, directly for the use of text segmentation. However, a similar task is performed using fuzzy systems by the authors of [1] whereby they instead combine Fuzzy systems and RST for text-summarisation. Authors of [13] demonstrate how one can use lexicalised syntax trees to illicit such information. The authors implement the use of dominance sets to achieve their overall goal of segmentation. The use of canonical lexical heads for a given sentence define the ROOT node of the subtree that word inhabits. The lexical head of the sentence derived using lexical head projection rules defined in [9], can be seen as the most salient word of that sentence i.e. the word that *carries* the sentence. This subtree can then be used to define the constituency for that sentence. We implement a very similar approach, instead, looking at the segments directly to determine the word that *carries* that segment the most. For each segment we can define, at most, two constituents encapsulate each segment. Finally, we can derive some overlap between each constituent, giving us our level of connectedness between the two segments.

2 Rationale

We are firstly motivated by the need for "gentler" segmentation algorithms. We look to employ a fuzzy system designed to take into account the cohesion, on a k-window-basis, applying a similar technique to that of text-tiling. Such a framework could allow us to generate more accurate probabilities for segmentation, whilst being able to easily expand, in terms of the potential for more variables and rules together with the problem size i.e. input text size. In smaller instances, probabilities calculations have less to account for meaning that outputs may seem more *erratic* due to the decreased fidelity of the values. However, in larger instances, the fidelity of the probabilities will increase allowing the benefits of the fuzzy system to shine. It is clear that for large instances of data, a heuristic approach may not be as accurate due to naturally higher levels of context and information needed to be considered. With more information present, there is a higher likelihood of new problem instances and occurrences that the current set of heuristics may not account for.

3 Methodology and Model Outline

This section covers the method and model used to fully actualize our fuzzy segmentation proposition. We begin by outlining the model and its data flow. We move secondly to the input variables necessary for the fuzzy system portion of the model. Thirdly, we explain the fuzzy system itself; justifying the fuzzy rules and setup used. We then define and resolve the edge cases present in the system; understanding the potential limitation they may incur. Finally, the implementation details are stated; explaining how we carried out the development of the model used.

3.1 Model

We describe a model similar to that of text-tiling. We can first theorize a single boundary between each word in a given text; including before the first word and after the last word (Leaving us with a set of $n + 1$ boundaries to query. We then define a window either side of a currently queried boundary; calculating 3 distinct variables: Syntactic coherence for the first and second window of words (denoted as coh_i and coh_j accordingly), and the syntactic incoherence between the two windows (denoted as $incoh$). These 3 variables are then fed into out fuzzy expert system whereby a set of rules are defined pertaining to the 3 values. The output of the system produces a probability as to the insertion of a boundary (denoted as p). If p is larger than a given threshold, we incur a boundary, otherwise, we do not. For each boundary or non-boundary returned per query, we log either a 1 or 0 accordingly. The model then moves to the next boundary in the set and repeats to aforementioned process. We are then left with a binary representation of the boundaries which is then fed as an input to our validation metrics; WindowDiff and WinPR. We now outline the definition and elicitation of each variable, together with alternative, considered methods.

Syntactic Cohesion Similar to the work presented in [3] with the C99 Algorithm for linear text segmentation, we also work with the concept of coherence when computing the connectedness of two or more words. Syntactic cohesion defines the level of linkage between two or more words or phrases. We calculate such a score, on a *shallow* basis, as a function of the path length from the target node to the first common ancestor (FCA) between the words or phrases in question. Currently, the score is directly proportionate to the path length between the leaf node/word and the FCA. Specifically, the score can be derived as follows:

$$score(leaf_i, leaf_j) = \frac{1}{max(depth(leaf_i), depth(leaf_j))} \tag{1}$$

$$depth(p, leaf_i) = |P_{height} - leaf_i^{height}| \tag{2}$$

$$p(leaf_i, leaf_j) = FCA(leaf_i, leaf_j) \tag{3}$$

Given the example in 1, if we are presented with two segments; $good, news, Dr$ and $Elmar, said$, we can determine, for each segment, the FCA between each words. This *distance*, described as the height between the "lowest" leaf and the the ancestor, measure to a degree, the connectedness within the segment.

Syntactic Incohesion and Phrase Structure Connectivity Building on the understanding provided by [3], we also choose to establish another, similar concept; *Syntactic incohesion*. We can define syntactic incohesion similarly to the way we define connectivity of each segment. Rather than calculating the cohesion between the two segments separately, we can instead calculate the cohesion of both segments combined; forming a segment-pair, where the set of words compared is equal to the $seg_i \cap seg_j$. When defining the incohesion in such a manner, we must take note of a few observations. The distance of the FCA between a larger assortment of words, likely to occur when the k-window is fairly large, leading to a segment of length $\leq 2k$, will be naturally larger than those of smaller sizes. If we define, artlessly, a constant that defines the level of incohesion based on this distance, we will find that large cohesive sentences will appear as incohesive due to the naturally large distance the words within the sentence would have to the FCA. Another observation is the potential for large sentences to have a FCA of the tree-root. In such cases, we can assume, as before, that those that share a root node for the FCA, can be considered largely incohesive. Syntactic inhcohesion plays a key role in tie-breaking when assessing the cohesions of segments i and j. Should the cohesion in both segments be high, there is still a chance that both segments could either be highly related or unrelated to one-another. Therefore, syntactic incohesions role is to determine the lack of connectedness between both segments to determining the production of a boundary.

Model Optimisations If we are faced with potentially a large k-size i.e. a large segment size, or we are presented with a large block of text to process

i.e. when we are working out the incohesion, determining the FCA for each segment or segment-pairs can prove to be a slow process, however, we find that it is unnecessary to calculate the FCA using each word. We can instead only calculate the FCA between the first and last word in each segment as the FCA will be equivalent to the largest FCA between all permutations. This allows for a consistent calculation time regardless of the k-size as we only need to utilise the first and last word of each segment per iteration.

Fuzzy System We have opted to use a fuzzy system (FS) that is both data and expert-knowledge driven. This was motivated by the fact that we can already illicit some initial data from the text tiling stage of our model. Rather than turn the problem, strictly, into a fuzzy problem, we have instead chosen to shape the FS and its rules to account for the present data elicited from the syntax. Other motivations include the need to move away from heuristic-based approaches i.e. syntactic rules as defined in [14] to govern segmentation as such a rigid rule-set can lead to poor adaptability especially when potentially trying to apply such work to languages that use different grammatical and syntactic structure. FSs can allow us to begin moving toward a more general-purpose segmentation model that has room to adapt and transform to accommodate new languages and salient syntactic features. The FS serves as a novel module to previous text tiling-based methods, with the outputs of the text-tiling portion, serving as the inputs of the FS. As stated above, we have opted to implement an expert-knowledge, data driven, model. With this means that our chosen rules are influenced by both the expert knowledge and the syntactic data elicited in the previous steps. For a given instance of the problem, we can define two segments of words.

3.2 Validation Metric

Window Diff We look to well established segmentation metric, derived from the work [2], called WindowDiff [11], used in the past by authors of [4] and [12] to validate their segmentation models with the latter proposing a new segmentation validation metric built off WindowDiff.

$$\frac{1}{N-k}\sum_{i=1}^{N-k}(|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})| > 0) \tag{4}$$

WindowDiff calculates the error between the automatically-generated boundaries, hyp and the ground truth boundaries ref. The original use case for this metric was to measure the accuracy of automatic sentence segmentation algorithms, however, we are looking into finer grain segmentation, beyond sentence boundaries. Therefore, rather than N denoting the number of sentences, it instead denotes the number of words within a target text window.

Second to WindowDiff, we also look into an alternative validation metric similar to WindowDiff, WinPR [12]. The authors motivation for genesis of this method stems from the added ability to produce a confusion matrix, and from

the matrix, derive a precision, accuracy an recall. We decide to use both in this instance for twofold: WindowDiff is well established in text segmentation, as such, should be considered first and foremost as the main metric to compare our model with others. Secondly, WinPR promises higher accuracies and fidelity in its metric; providing a higher level of analysis during the decomposition of our method when it comes time to explain its intricacies and capabilities in parsing and segmentation.

WinPR Derived from WindowDiff, the authors of [12] have proposed an improved version of WindowDiff. With their method, comes, importantly, the ability to precisely compare our method with that of similar work. More specifically, WinPR allow us to achieve such comparison with the ability to derive a confusion matrix for the calculated segmentation – allowing for the subsequent derivation of precision, recall, accuracy; variables which our *rival* methods used to assess their ability in segmentations. We can calculate our confusion matrix as follows:

$$\begin{aligned} TP &= \sum_{i=1-k}^{n} min(ref_{i,i+k}, hyp_{i,i+k}) \\ TN &= -k(k-1) + \sum_{i=1-k}^{n} (k - max(ref_{i,i+k}, hyp_{i,i+k})) \\ FP &= \sum_{i=1-k}^{n} max(0, hyp_{i,i+k} - ref_{i,i+k}) \\ FN &= \sum_{i=1-k}^{n} max(0, ref_{i+i-k} - hyp_{i,i+k}) \end{aligned} \tag{5}$$

where *ref* is the reference segmentation and *hyp* is the computed segmentation. Segmentations can be abstracted as an index-able binary list where 1's denote a boundary and 0 denotes no boundary.

4 Conclusion and Further Work

The proposition of fuzzy boundaries has shown to potentially provide a general-purpose framework that can be adapted for multiple use cases. Currently, the ground truths to validate this approach are not openly available instead, we can look to producing the ground truths with a similar segmentation model, SLSeg, that has performed exceptionally well when segmenting text similar to ours. With the fuzzy system comes a large expanse for customisation is terms of rule and input tweaking to provide a multifaceted solution to many segmentation problem-instances i.e. sentence and paragraph segmentation. Furthermore, the chosen validation metrics allow for straight-forward comparability between other existing segmentation models – allowing for clear indications as to the versatility of the model for any chosen use case.

References

1. Ayed, A.B., Biskri, I., Meunier, J.G.: Automatic text summarization: A new hybrid model based on vector space modelling, fuzzy logic and rhetorical structure analysis. In: International Conference on Computational Collective Intelligence. pp. 26–34. Springer (2019)
2. Beeferman, D., Berger, A., Lafferty, J.: Statistical models for text segmentation. Machine learning **34**(1-3), 177–210 (1999)
3. Choi, F.Y.: Advances in domain independent linear text segmentation. arXiv preprint cs/0003083 (2000)
4. Dias, G., Alves, E., Lopes, J.G.P.: Topic segmentation algorithms for text summarization and passage retrieval: an exhaustive evaluation. In: AAAI. vol. 7, pp. 1334–1340 (2007)
5. Hearst, M.A.: Texttiling: A quantitative approach to discourse segmentation. Tech. rep., Citeseer (1993)
6. Le Thanh, H., Abeysinghe, G., Huyck, C.: Automated discourse segmentation by syntactic information and cue phrases. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), Innsbruck, Austria. pp. 411–415 (2004)
7. Li, J., Sun, A., Joty, S.R.: Segbot: A generic neural text segmentation model with pointer network. In: IJCAI. pp. 4166–4172 (2018)
8. Louis, A., Nenkova, A.: A coherence model based on syntactic patterns. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1157–1168 (2012)
9. Magerman, D.M.: Statistical decision-tree models for parsing. In: Proceedings of the 33rd annual meeting on Association for Computational Linguistics. pp. 276–283. Association for Computational Linguistics (1995)
10. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text **8**(3), 243–281 (1988)
11. Pevzner, L., Hearst, M.A.: A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics **28**(1), 19–36 (2002)
12. Scaiano, M., Inkpen, D.: Getting more from segmentation evaluation. In: Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies. pp. 362–366 (2012)
13. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics. pp. 228–235 (2003)
14. Tofiloski, M., Brooke, J., Taboada, M.: A syntactic and lexical-based discourse segmenter. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. pp. 77–80 (2009)
15. Wan, S., Paris, C.: Experimenting with clause segmentation for text summarization. In: TAC. Citeseer (2008)