C-HTS: A Concept-based Hierarchical Text Segmentation approach

Mostafa Bayomi, Séamus Lawless
ADAPT Centre, Knowledge and Data Engineering Group,
School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland
bayomim@tcd.ie, seamus.lawless@scss.tcd.ie

Abstract

Hierarchical Text Segmentation is the task of building a hierarchical structure out of text to reflect its sub-topic hierarchy. Current text segmentation approaches are based upon using lexical and/or syntactic similarity to identify the coherent segments of text. However, the relationship between segments may be semantic, rather than lexical or syntactic. In this paper we propose *C-HTS*, a Concept-based Hierarchical Text Segmentation approach that uses the semantic relatedness between text constituents. In this approach, we use the explicit semantic representation of text, a method that replaces keyword-based text representation with concept-based features, automatically extracted from massive human knowledge repositories such as Wikipedia. *C-HTS* represents the meaning of a piece of text as a weighted vector of knowledge concepts, in order to reason about text. We evaluate the performance of *C-HTS* on two publicly available datasets. The results show that *C-HTS* compares favourably with previous state-of-the-art approaches. As Wikipedia is continuously growing, we measured the impact of its growth on segmentation performance. We used three different snapshots of Wikipedia from different years in order to achieve this. The experimental results show that an increase in the size of the knowledge base leads, on average, to greater improvements in hierarchical text segmentation.

Keywords: Hierarchical Text Segmentation, Explicit Semantic Analysis, Semantic Relatedness, Wikipedia

1. Introduction

Text segmentation aims to divide text into coherent segments which reflect the sub-topic structure of the text. It is widely used as a pre-processing task for Information Retrieval (Prince and Labadié, 2007) and several Natural Language Processing (NLP) tasks such as Text Summarization (Boguraev and Neff, 2000) and Question Answering (Tellex et al., 2003). Text segmentation is essentially based on measuring the coherence between atomic units of text (sentences or paragraphs). Some approaches use lexical similarity, which compares units of text that are represented as vectors of the same, or similar, words (Choi, 2000). These approaches are limited, in that they rely on endogenous knowledge extracted from the documents themselves. Relying on such knowledge does not reveal much about the meaning beyond the text.

Some approaches started to enrich the text representation by exploiting its semantic meaning by using the Latent Semantic Analysis (LSA) (Choi et al., 2001). However, these approaches require a very large corpus, and consequently the pre-processing effort required is significant. On the other hand, some other approaches started to use external resources such as WordNet to enrich text (Stokes et al., 2004). However, such resources cover only a small fragment of the language lexicon. Furthermore, the use of such lexical resources offers little knowledge about the different word representations.

Buchanan and Feigenbaum (1982) stated that: "*The power of an intelligent program to perform its task well depends primarily on the quantity and quality of knowledge it has about that task.*" Hence, in this research, we are trying to enrich the text representation by replacing traditional text representation methods with a concept-based representation that exploits an external knowledge base to reveal more knowledge about the text.

When a person reads a text, the eyes read the words (the lexical representation of text) and send these words to the human's cognitive system, the brain. The brain starts to make sense of these words based on the knowledge of the reader. For example, the name "*Albert Einstein*" in a text document is read by the eyes and then sent to the brain, which starts to map the name to the different concepts that the person knows about Einstein such as: "*Theory of Relativity*", "*Physics*", "*Nobel Prize*", etc. The information that the brain maps the name to, is dependent upon how much knowledge this person has. If the individual does not know about Einstein, the brain would make no sense of that name. The individual could potentially ask other people who have different collections of knowledge for assistance, creating an intellectual representation through collaboration. In this research, we are trying to recreate this methodology in a segmentation algorithm. It is our contention that using this approach to understand text would make a more accurate approach to text segmentation.

The essential task in any text segmentation algorithm is to measure the coherence between two adjacent text blocks. Being inherently limited to lexical representation, current approaches cannot reveal much about the coherence between text blocks. Consider the following two sentences for example:

- Albert Einstein is a German scientist who was born on the 14th of March 1879.
- Mileva Marić was born on December 19, 1875 into a wealthy family in Titel, Serbia.

Lexically, the two sentences are not similar because both have different names, cities and dates. For a segmentation approach that solely relies upon a lexical representation of text, the two sentences are not similar or even related to each other. Even for an approach that uses a learning model to learn text representation, if it has not seen the entities mentioned in the sentences together in a training set it will be difficult for it to infer the relation between the two sentences. In fact, Mileva Marić is Einstein's ex-wife, they both worked in physics and they had three children. Hence, an ideal approach to reveal such information about the two sentences, and to measure their relatedness, would use the explicit semantic representation of text based on a knowledge base. Such a knowledge base should be based on human cognition and be intuitive to use and reason over, with no limits on domain coverage or conceptual granularity. Creating and maintaining such knowledge base requires enormous effort on the part of many people. Luckily, such a collection already exists in the form of

Wikipedia, which is one of the largest knowledge repositories on the Web. Hence, relying on such human-organised intensive knowledge reveals more meaning of the text that we want to segment regardless of the approach (linear or hierarchical) or the algorithm that we use for segmentation.

In this paper we propose *C-HTS*, a Concept-based Hierarchical Text Segmentation approach that uses semantic representation to measure the relatedness between text blocks and then builds a tree-like hierarchy of the document to reveal its semantic structure. *C-HTS* capitalises on the human knowledge encoded in Wikipedia and uses its concepts to leverage information about text that cannot be deduced solely from the input texts being processed. We evaluate *C-HTS* on two hierarchical datasets and we compare its performance against selected state-of-the-art approaches. We also use different snapshots from Wikipedia to assess the influences of knowledge base size on the hierarchical segmentation task.

The contributions of this paper are threefold. First, we propose a new approach to hierarchical text segmentation that uses explicit semantic representation of text as a substitute for traditional lexical representation. Second, we assess the impact of knowledge base size on the segmentation task through an experiment where we use different snapshots of Wikipedia from different years. Third, we have processed a recent Wikipedia snapshot (April 2017) as described in (Gabrilovich & Markovitch 2009) to use as the underlying concept space for the explicit semantic analysis of text. This processed Wikipedia snapshot[1]s along with a Java implementation of *C-HTS*[2] are publicly available.

2. Related Work

Natural Language Processing (NLP) has different tasks (Lawless et al. 2015; Bayomi et al. 2016; Naili et al. 2016). One of these tasks is Text Segmentation that is considered an essential task for other NLP tasks (Boguraev & Neff 2000). Text Segmentation can be classified into two main broad classes: Linear and Hierarchical text segmentation. Linear text segmentation approaches focus on segmenting text into coherent segments where each segment represents a specific topic (Choi 2000). An early linear text segmentation algorithm was the TextTiling approach introduced by Hearst (1997). TextTiling applied linear text segmentation by measuring the lexical similarity between text blocks. Text blocks are the smallest units that constitute the text. They range from one sentence (Ye et al. 2008) to multiple sentences (paragraphs) (Bayomi et al. 2015). TextTiling is a content-based text segmentation algorithm that uses a sliding window to segment a text. The calculation is accomplished using two vectors containing the number of terms occurring in each block. Utiyama and Isahara (2001) proposed a linear approach, U00, that is based on language models, where they use dynamic programming and the probability distribution of words to rank and select the best segments. Eisenstein and Barzilay (2008) proposed a Bayesian approach to unsupervised topic segmentation. They showed that lexical cohesion between text segments can be placed in a Bayesian context by modelling the words in each topic segment. Galley et al. (2003) proposed *LcSeg*, a *TextTiling*-based algorithm that uses *tf-idf* term weights, which improved the text segmentation results. Another well-known linear text segmentation algorithm is *C99*, introduced by Choi (2000). *C99* segments a text by combining a rank matrix, transformed from the sentence-similarity matrix, and divisive clustering.

Hierarchical text segmentation, on the other hand, focuses on discovering more fine grained subtopic structures in texts (Kazantseva & Szpakowicz 2014). An early hierarchical text segmentation approach was proposed by Yaari (1997). Yaari used paragraphs as the elementary units for his algorithm and he measured the cohesion between them using cosine similarity. An agglomerative clustering approach is then applied to induce a dendrogram over paragraphs. Eisenstein (2009) proposed a hierarchical Bayesian algorithm based on Latent Dirichlet Allocation (LDA). Eisenstein modelled each word token as a draw from a pyramid of latent topic models to create topical trees.

Both, hierarchical and linear approaches attempt to place boundaries between utterances. There are three main approaches to detect boundaries within text (Prince & Labadié 2007):

1) *Similarity-based methods*: where text blocks are represented as vectors of their terms and then a measure is used to find the proximity by using (most of the time) the cosine of the angle between these vectors. For example, C99 (Choi 2000) uses a similarity matrix to generate a local classification of sentences and isolate topical segments.

2) *Graphical methods*: a representation of term frequencies is plotted on a graph to identify topical segments (which are dense dot clouds on the graph). The DotPlotting algorithm (Reynar 1994) is the most common example of the use of a graphical approach of text segmentation.

3) *Lexical chain-based methods*: the notion behind lexical chains is to chain semantically related words together via a thesaurus. It was proposed by Morris and Hirst (1991). Multiple occurrences of a term in a document are linked together through a chain. This chain is considered broken when there are too many sentences between two occurrences of a term. Segmenter (Kan et al. 1998) is an example of approaches that use lexical chains in text segmentation. It uses lexical chains with a subtle adjustment as it determines the number of necessary sentences to break a chain in function of the syntactical category of the term.

All these approaches rely upon the traditional bag-of-words representation of text. However, a representation based solely on the endogenous knowledge in the documents themselves does not reveal much about the meaning of the text. Hence, some approaches started to enrich the text representation by exploiting its semantic meaning. Choi et al. (2001) enriched their approach, C99, by using the Latent Semantic Analysis (LSA). They applied latent concept modelling to the similarity metric. They proved that using

[1] Wikipedia 2017 snapshot processed for ESA: https://goo.gl/JZhEvm,

[2] https://github.com/bayomim/C-HTS

LSA improved the quality of their segmenter. However, these LSA-based approaches require a very large corpus, and consequently the pre-processing effort required is significant.

Some other approaches started, on the other hand, to use external resources to enrich text. Stokes et al. (2004) proposed a new approach, SeLeCT, that uses the WordNet thesaurus as an external lexical resource to add semantic links between words to create lexical chains from these links with respect to a set of chain membership rules. However, the use of such lexical resources offers little information about the different word representations. Furthermore, such resources cover only a small fragment of the language lexicon.

Ontologies have been widely used in different tasks to give a conceptual representation of entities (Bayomi & Lawless 2016). Recently, some approaches have emerged that segment text by exploiting the conceptual representation of its constituent terms. For example, we proposed OntoSeg (Bayomi et al. 2015) as a hierarchical text segmentation approach that is based on the ontological similarity between text blocks. The approach annotates text using a named entity recognition algorithm and text entities are extracted. The extracted entities are then mapped to their concepts (classes) from an ontology (DBpedia in the experiments). The sentences in the text are then represented as vectors of concepts from the ontology. The similarity between two text blocks (one or more sentences) is measured based on the similarity between the concepts of their entities in the ontology using the *is-a* relation. Naili et al. (2016) integrated a domain ontology in the topic segmentation in order to add external semantic knowledge to the segmentation process. They proposed two topic segmenters called TSS-Ont and TSB-Ont based on C99 and TextTiling respectively. They used the same techniques as C99 and TextTiling but replaced lexical similarity with concept similarity.

Although these approaches relied on an external resource and used an ontology to add a semantic layer to the segmentation process, they suffer from some drawbacks, such as: they solely extract named entities from text, and in a text with few entities or with poor performance from the named entity extraction algorithm, measuring the similarity between text blocks is not feasible. Furthermore, these approaches measure the semantic similarity between entities rather than the semantic relatedness. As argued by Budanitsky and Hirst (2006), relatedness is more general than similarity. Furthermore, dissimilar entities may also be semantically related by other relationships such as meronymy, antonymy, functional relationship or frequent association.

In this paper we propose C-HTS, a hierarchical model of text segmentation that uses the semantic relatedness between text blocks to produce a tree-like structure of a text document. C-HTS uses the explicit semantic representation of text to measure how text blocks are semantically related based on concepts from a knowledge base. C-HTS uses the exogenous knowledge (externally supplied), rather than the endogenous knowledge extracted from the documents themselves. The approach uses Wikipedia as an external knowledge base to enrich the text representation in a very high-dimensional space of concepts.

The purpose of measuring semantic relatedness is to allow computers to reason about text. Various approaches have been proposed in the literature to measure the semantic relatedness between terms using an external knowledge source. Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch 2007) is a technique that provides a semantic representation of text in a space of concepts derived from Wikipedia. ESA defines concepts from Wikipedia articles e.g., BARACK OBAMA and ACOMPUTER SCIENCE. A target term is essentially represented as a vector of concepts in Wikipedia based on how this term is mentioned in the concept's article. Relatedness is then calculated as the cosine similarity between the two vectors of the target terms. Another approach that uses the link structure of Wikipedia to measure semantic relatedness is the Wikipedia Link-based Measure (WLM) (Witten & Milne 2008). WLM measures the relatedness between two terms using the links found within their corresponding Wikipedia articles rather than using the articles' textual content. In this research we use Explicit Semantic Analysis (ESA) and we use Wikipedia as its source of knowledge. ESA has been widely used in a variety of tasks such as concept-based information retrieval (Egozi et al. 2011) and text classification (Chang et al. 2008) among other tasks. The efficacy of ESA has been proven compared to other approaches that do not rely on explicit knowledge bases.

3. Semantic Relatedness Using Wikipedia

The core idea of our algorithm is the use of an external knowledge base to enrich text representation to measure the semantic relatedness between terms, and thus sentences, and to utilise this in hierarchical text segmentation. The notion behind using explicit semantic relatedness is that it relies on a knowledge base that is built and continuously maintained by humans. The knowledge base that we use in this research is Wikipedia, the largest and fastest growing encyclopaedia in existence. This knowledge base is considered a collaborative effort that combines the knowledge of hundreds of thousands of people. Many approaches have exploited Wikipedia to measure the semantic relatedness between terms. In this research, we use Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch 2007) for this task. ESA is a method that represents meaning in a high-dimensional space of concepts, automatically driven from human-built knowledge repositories such as Wikipedia.

ESA maps a term to a concept vector, this vector contains the term's association strengths to concepts in Wikipedia. A concept is a Wikipedia article (e.g. ALBERT EINSTEIN). This concept is represented as a vector of terms which occur in that article weighted by their *tf-idf* score. After generating terms from the concept article, an inverted index is created that maps each term to a list of concepts in which this term appears. The name, Explicit Semantic Analysis, stems from the way vectors are comprised of concepts that are manually defined, as opposed to the mathematically derived contexts used by Latent Semantic Analysis.

Each input term in a text processing task (e.g. segmentation) can be represented as a vector of concepts that the term is associated with, accompanied by the degree of association between the term and each concept. The semantic relatedness between two given terms is measured by computing the cosine similarity between the concept vectors of the two terms. For larger text fragments (sentence or paragraph), a concept vector is retrieved for each term in the fragment, then the semantic relatedness between two text fragments is measured by computing the cosine similarity between the centroid of the vectors representing the two fragments.

To elaborate on the notion of the semantic relatedness using ESA, consider the two sentences in the example mentioned in Section 1. Each term in each sentence is mapped to a vector of concepts from the vector space. Each sentence is then represented as the centroid of the vectors of the sentence's terms. For the first sentence, the centroid of the vectors contains the following concepts (among other concepts):

- ALBERT EINSTEIN AWARD
- THE EVOLUTION OF PHYSICS
- HANS ALBERT EINSTEIN → (*second child and first son of Albert Einstein and Mileva Marić*)
- ELSA EINSTEIN → (*the second wife of Einstein*)

And the centroid of the vectors of the second sentence contains the following (among other concepts):

- MILEVA MARIĆ
- HANS ALBERT EINSTEIN
- ELSA EINSTEIN
- EINSTEIN FAMILY

From these vectors, we can see that the two sentences have concepts in common. This shows that although the two sentences are not lexically similar, they are semantically related to each other.

4. The C-HTS Algorithm

4.1 C-HTS Components

The *C-HTS* algorithm proposed by this research consists of three phases:

4.1.1 Morphological Analysis

In this phase, the target is processed to be split it into sentences and remove stopwords as they are generally assumed to be of less, or no, informational value. The remaining words are then stemmed and converted to their root. In this research we use the Porter stemmer (Porter, 1980). This morphological analysis technique has been used in processing the Wikipedia terms and concepts while building the concept space from Wikipedia. The remaining terms are then used as input for the next phase.

4.1.2 Calculating the Semantic Relatedness

The key idea in *C-HTS* consists of treating the segmentation of text as an examination of the semantic relatedness between text blocks rather than traditional lexical similarity. A text block is the elementary unit of the segmentation algorithm, which is one sentence in *C-HTS*. For each sentence, and for each term in that sentence, the term is mapped to a vector of concepts from the concept space that was created from Wikipedia. The semantic relatedness between two (adjacent) sentences is calculated as the cosine similarity between the centroid of the vectors representing the individual terms in each sentence.

4.1.3 Hierarchical Agglomerative Clustering

C-HTS is an iterative approach that uses the bottom-up Hierarchical Agglomerative Clustering (HAC) algorithm for text segmentation. Hierarchical clustering algorithms have been studied extensively in the clustering literature (Jain and Dubes 1988). A typical agglomerative clustering algorithm successively merges documents into clusters based on a specific criterion such as their similarity with one another. In *C-HTS* we transfer the agglomerative clustering technique from document level to text level. The clustering process is done in *C-HTS* between text blocks within one document as opposed to across documents. The main topic for research in the HAC algorithm is the proximity test. In *C-HTS*, we apply the semantic relatedness between text blocks as the proximity test.

When applying hierarchical agglomerative clustering on text blocks the algorithm successively agglomerates blocks that are deemed to be semantically related to each other, thus forming a text structure. *C-HTS* uses HAC because it is a bottom-up clustering approach. The idea behind using a bottom-up approach in text segmentation is that it starts from the smallest clusters, that are considered the seeds of the text, and then builds the text structure by successively merging the semantically coherent clusters. This way of building the document structure can be regarded as a hierarchically coherent tree that is useful to support a variety of search methods as it provides different levels of granularity for the underlying content.

Conceptually, applying the HAC algorithm on text blocks produces a hierarchy tree or a dendrogram. In this tree, the leaf nodes correspond to individual blocks (sentences). When two blocks are merged together, a new node is created in this tree corresponding to this larger merged group. Figure 1 shows the resulting dendrogram from *C-HTS* for a sample text. In the dendrogram depicted in Figure 1, we can see that for each iteration of *C-HTS* a new level (horizontal dotted lines) is constructed from the agglomeration process on the previous level. Each level is considered a different representation of the document granularity. The level of granularity increases as we move from the root to the bottom of the tree (the leaves). For example, in level 5 in the dendrogram, we can see that the document at that level of granularity can be segmented into two segments with boundaries 19 & 25.

4.2 Complexity analysis

Hierarchical agglomerative clustering algorithms for clustering text documents in general takes an order of $O(N^2)$ steps. This is because at each stage in the algorithm the proximity of the newly merged object to all other available segments is computed. On the other hand, in *C-HTS*, we apply the hierarchical agglomerative clustering on text level. Since we need to preserve the linear order in text, we only compute the proximity between a cluster and its surrounding clusters. Hence, *C-HTS* takes an order of *O(N)* steps on text level.

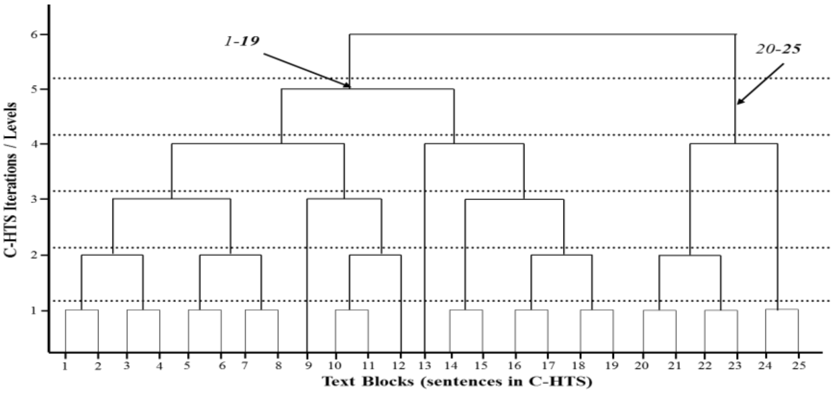

Figure 1 C-HTS output as a dendrogram of a sample text

4.3 Word Sense Disambiguation

Word Sense Disambiguation (WSD) is the task of identifying the meaning of a term, when the term has multiple meanings, based upon the context of where it appears (Navigli 2009). For example, “light” can mean “not heavy” or “illumination”, what identifies its meaning is the context of where “light” is used. For a natural language processing task like text segmentation, disambiguating such words would allow the task to better understand the meaning of the sentence and to reason about it and thus enhance the quality of the segmentation. For lexical segmenters, being inherently limited to lexical representation of text, these approaches require an extra level of sophistication to disambiguate words.

In *C-HTS*, we measure the relatedness between sentences as the cosine similarity between the centroid of the vectors representing the two sentences. This interpretation of text is considered an implicit disambiguation of terms. For example, a sentence that has the term “*Apple*” amongst other computer related terms, taking the centroid of the vectors will boost the computer-related concepts and will disambiguate the term effectively. To illustrate how words are disambiguated in *C-HTS*, consider the following sentence: “*I love fruit, particularly a nice apple*”. In this sentence, after applying morphological analysis (Section 4.1.1), the remaining prominent terms are *love*, *fruit*, *particularli*, *nice* and *appl*. Among these terms, the word “apple” has different interpretations. From the underlying concept space that we have created from Wikipedia, the top concepts generated for the word *apple* are:

- APPLE DAY → (*related to apple fruit*)
- APPLE SPECIALIST → (*related to apple Inc.*)
- APPLE EXTENDED KEYBOARD → (*related to apple Inc.*)
- EMPIRE (APPLE) → (*related to apple fruit*)
- APPLE STORE (ONLINE) → (*related to apple Inc.*)

These concepts are mostly talking about the company, Apple Inc. When considering the centroid of the vectors representing this sentence, the top generated concepts are (among others):

- FRUIT PICKING
- ROME APPLE → (*a kind of apple originating near Rome Township, Ohio*)
- LIST OF APPLE CULTIVARS
- EMPIRE (APPLE) → (*a kind of apple derived from a seed grown in 1945*)

From these concepts, we can see that they all are related to the fruit apple. This proves that considering the centroid of the vectors of a sentence disambiguates the terms without adding extra sophisticated text processing layers. This vector can also be seen as a representation of the context of that sentence. This in turn enhances the understandability of text and enhances the segmentation quality.

5. C-HTS Evaluation

Research on hierarchical text segmentation has been scarce and most state-of-the-art approaches evaluated their hierarchical approaches on linear segmentation datasets. For example, Yaari (1997) evaluated his approach on the *Stargazers* article. He compared his approach against a linear text segmentation approach, TextTiling. OntoSeg (Bayomi et al. 2015) was evaluated using Choi’s dataset for linear text segmentation evaluation. Evaluating a hierarchical text segmentation algorithm using a linear dataset does not give a realistic picture of the performance of the hierarchical algorithm. This is because the output of a hierarchical algorithm is a tree structure, while a linear dataset has consequently segmented chunks of text. Hence, selecting an appropriate dataset is a critical step in the evaluation process.

5.1 Datasets

In this research, we argue that *C-HTS* is applying the hierarchical text segmentation as if a human would perform the task. To prove this assumption, a gold standard dataset that is created by humans is needed. Furthermore, the dataset needs to be suitable for a hierarchical text segmentation task. Luckily, Kazantseva and Szpakowicz (2014) proposed two datasets that are suitable for evaluating hierarchical text segmentation and both were annotated by humans. The authors evaluated their approach, Hierarchical Affinity Propagation for Segmentation (HAPS), against two well-defined datasets: the *Moonstone* dataset and the Wikipedia dataset compiled by Carroll (2010).

Moonstone dataset: This dataset consists of nine chapters of the Moonstone novel. Kazantseva and Szpakowicz (2014) employed human annotators to annotate the dataset and to identify the hierarchical structure of each text document. The annotators were asked to read a chapter and split it into top-level segments according to where they can see a shift in topic. Each chapter was annotated by 3-6 people (4.8 on average)[3].

Wikipedia dataset: This dataset was compiled by Carroll (2010). The dataset consists of 66 Wikipedia articles on various topics. The html pages were converted to flat text, and unneeded content such as navigation boxes, and image captions were removed. The hierarchical structure for each article is created automatically from the structure of the Wikipedia page, i.e. heading text was replaced with a boundary marker, indicating the heading depth. This depth represents the level in the text's hierarchical structure. While the levels in the Wikipedia dataset were created automatically, the original structure of the documents is created by the human authors who contribute to Wikipedia. Thus it is considered a human annotated dataset.

Since *C-HTS* is based on the external knowledge base to enrich text representation, evaluating it on these two datasets will give us a realistic picture of the performance of *C-HTS* as a concept based approach. This is due to the inherent human involvement in the construction process of the two datasets.

5.2 Baselines

To evaluate the quality of segmentations produced by *C-HTS*, there is a need to compare its performance against hierarchical text segmentation approaches. Work on hierarchical text segmentation has been scarce. To the best of our knowledge, the only publicly available hierarchical segmenter (along with a dataset) is HAPS that was proposed by Kazantseva and Szpakowicz (2014). HAPS[4] is a hierarchical text segmentation approach that is based on a graphical model for hierarchical clustering called Hierarchical Affinity Propagation (Givoni et al. 2011). The input for HAPS is a matrix of similarity between text blocks. HAPS requires the desired number of levels to be in the produced topical tree and a preference value for each data point and each level. HAPS also finds a centre for each segment at every level of the produced topical tree, a data point which best describes the segment.

HAPS was compared against two linear segmenters *MCSeg* (Minimum Cut Segmenter) (Malioutov & Barzilay 2006) and *BSeg* (Bayesian based Segmenter) (Eisenstein 2009). These two systems were chosen because they are representative of the existing text segmentation methods, and their implementations are freely available on the internet. *MCSeg* casts text segmentation in a graph-theoretic framework. In this approach, text is abstracted into a weighted undirected graph, where the nodes of the graph correspond to text blocks and edge weights represent the pairwise block similarity. Text segmentation in *MCSeg* corresponds to a graph partitioning that optimises the normalised-cut criterion. In *BSeg*, the lexical cohesion between segments is placed in a Bayesian context. The words are modelled in each topic segment as draws from a multinomial language model associated with the segment.

To obtain hierarchical segmentation from these two linear segmentation systems, both systems were run first to produce top-level segmentations. Each segment thus computed was a new input document for segmentation. The procedure was repeated twice to obtain a three level structure of the text.

In this research we compare our approach, *C-HTS*, with *HAPS* and the other two baselines proposed by Kazantseva and Szpakowicz. For evaluation consistency, we use their experimental settings by evaluating the top three levels (excluding the root) of the document structure produced by *C-HTS*.

5.3 Evaluation Metric

We evaluate *C-HTS* using the well-known metric *windowDiff* (Pevzner & Hearst 2002). *windowDiff* is a penalty measurement metric, which means that lower scores indicate higher segmentation accuracy. *windowDiff* was proposed by Pevzner and Hearst as a modification to the *Pk* evaluation metric proposed by Beeferman et al. (1997). *windowDiff* is computed by sliding a window across the input sequence and at each step examining whether the hypothesised segmentation is correct about the separation (or not) of the two ends of the window. It counts the difference of the number of segment boundaries in the given window between the two partitions. *windowDiff* is defined as:

$$windowDiff\ (ref\ ,hyp) = \frac{1}{K-k}\sum_{i=1}^{K-k}(|b(ref_i,ref_{i+k}) - b(hyp_i,hyp_{i+k})| > 0)$$

where *ref* is the correct segmentation for reference, *hyp* is the segmentation produced by the model, *K* is the number of sentences in the text, *k* is the size of the sliding window and *b(i, j)* is the number of boundaries between sentences *i* and *j*.

windowDiff is designed to evaluate linear text segmentation not hierarchical trees. Hence, in our evaluation, and for the sake of comparability we follow the same technique as Kazantseva and Szpakowicz (2014). Each level of the text hierarchy is treated as a separate segmentation and each hypothetical level is compared against a corresponding level in the reference segmentation.

5.4 Results

The *Moonstone* dataset has on average 4.8 annotations per chapter. To obtain a realistic picture of the results across the different annotators per file, each hypothetical segmentation is separately compared against each available gold standard. After that, the averages across all annotators are taken as the final score. For the two datasets, Table 1 shows results of the comparison between *C-HTS* and the other three baselines using the *windowDiff* evaluation metric. Since *C-HTS* and *HAPS* are inherent hierarchical text segmentation approaches, both were run without knowing the number of segments. *BSeg* was able to run with and without that parameter. In our results, we report the *BSeg* run without this parameter. *MCSeg*, on the other hand, required that the exact number of segments to be specified. This makes it considerably more informed than others.

The results show that *C-HTS* performs well on both datasets compared to the baselines, even when compared to more informed baseline. For the Wikipedia dataset, *C-HTS* performs better than the baselines on all three levels.

[3] For more details about the *Moonstone* dataset, the reader is referred to (Kazantseva and Szpakowicz, 2014).

[4] HAPS implementation and the Moonstone dataset are available here: https://github.com/anna-ka/HAPS

This proves that using the explicit semantic representation of text gives more understanding of the meaning of the text, and thus enhances the process of hierarchical text segmentation.

	Level	*Moonstone windowDiff*	*Wikipedia windowDiff*
C-HTS	3 (top)	**0.320**	**0.360**
	2 (middle)	0.507	**0.400**
	1 (bottom)	**0.488**	**0.409**
HAPS	3 (top)	0.337	0.421
	2 (middle)	**0.422**	0.447
	1 (bottom)	0.556	0.617
MCSeg	3 (top)	0.375	0.440
	2 (middle)	0.541	0.424
	1 (bottom)	0.601	0.471
BSeg	3 (top)	0.600	0.637
	2 (middle)	0.447	0.877
	1 (bottom)	0.545	0.952

Table 1. Evaluation of *C-HTS*, HAPS and iterative versions of MCseg and BSeg using *windowdiff* per level

For the Moonstone dataset, *C-HTS* performs favourably on the top and bottom levels but we notice that its performance on the middle level is not better than HAPS and BSeg. We argue that this is because in the Moonstone dataset the boundary for each level, in each document, was placed by a number of different annotators, hence, there can be mixed agreement between those annotators on the correct placement of the level boundary. On the other hand, in Wikipedia dataset, the original article hierarchy (where levels are obtained from) was created and updated with the agreement of the Wikipedia article contributors.

6. Discussion

6.1 Elementary units for the segmenter

The bottom-up hierarchical text segmentation algorithms start with atomic text pieces as their elementary units and then successively grow areas of coherence at the most appropriate place. The elementary units could be of a fixed size, such as a specific number of sentences, or could be of a mutable size such as paragraphs. For example, Yaari (1997) and Kazantseva & Szpakowicz (2014) used paragraphs as the elementary units for their segmenters, while in *C-HTS* we use one sentence as the elementary unit.

The size of the elementary units is an influential parameter for the segmentation algorithm and it has implications on the segmentation accuracy. Previously, we experimented with the influence of different elementary unit sizes on the hierarchical segmentation task (Bayomi et al. 2015). We experimented with sizes ranging from one to four sentences per unit. The best run we reported in our experiments was when we used one sentence as the elementary unit. The results also concluded that the higher the size of the elementary unit, the lower the accuracy of the segmentation.

This also adds to the understanding of the inconsistency of *C-HTS* performance on the Moonstone dataset. Besides the disagreement between the human annotators about the correct placement of level boundary, the elementary units presented to the annotators, to build the gold standard, were paragraphs. As a result, and for the evaluation consistency, we had to set the elementary units for *C-HITS* to be paragraphs which impacted the performance of the algorithm. This can be seen in the results of the first experiment (and the following experiment) where the performance of *C-HTS* on the Wikipedia dataset, where we use one sentence as the elementary unit, gives, on average, lower error rates than its performance on the Moonstone dataset.

6.2 Text Granularity

Hierarchical text segmentation approaches produce a structural representation of text that represents different levels of granularity. In *HAPS*, the desired number of levels needs to be passed as a parameter to the algorithm. In contrast, in *C-HTS*, it does not need to know number of levels that are needed in the output structure because the structure produced by *C-HTS* depends on the coherence between the atomic units of the text. This way of building the structure makes the output more granular and facilitates its use in different tasks like Information Retrieval. Identifying the number of levels of the output limits the usage of the produced hierarchy, as each task requires a different level of granularity. Hence, from this point of view, *HAPS* is considered a task-dependent approach, as its parameters need to be set depending on the task in question. On the other hand, *C-HTS* is considered a task-independent approach as it produces all the available levels of granularity in the processed document, hence it can be used with different tasks.

6.3 Multilingual C-HTS

C-HTS is based on the concept space built from Wikipedia. Wikipedia is the largest encyclopaedia in existence that is available in dozens of languages. Building a concept space for these languages would help an ESA-based task to be used with texts in different languages. Gurevych et al. (2007) applied ESA to the German-language Wikipedia and used it for semantic relatedness and information retrieval tasks. Their experiments showed that using ESA was superior compared to a system based on the German version of WordNet (GermaNet).

The core of *C-HTS* is the process of measuring the semantic relatedness between clusters using the explicit semantic interpretation of text. This process is essentially based on the underlying concept space that we have built from Wikipedia. Moving *C-HTS* from one language to another can be done easily. Changing the language of the underlying concept space would make no difference in the running process of *C-HTS*. The only step which must be changed is the morphological analysis to filter out the prominent terms in text. This step is relatively easy to implement as there has been a large volume of work completed on morphological analysis for languages other than English (Rafferty & Manning 2008). Hence, *C-HTS* can be seen as a multilingual hierarchical text segmentation approach that can semantically represent text and reason about it regardless the language of the text.

7. The Impact of Knowledge Breadth

In this research, we use a concept space that is built from the text of a knowledge base articles (Wikipedia). Anderka and Stein (2009) showed that the nature of the text collection used to build the concept space has much less impact on the explicit semantic analysis performance than its size.

Wikipedia is being constantly expanded and updated by different contributors who add new articles and extend the existing ones. Consequently, the amount of knowledge in Wikipedia is expanding. We conjecture that such expansion, and the growth of information available in the knowledge base should impact the accuracy of the segmentation process. To test this assumption, we acquired different snapshots of the entire Wikipedia knowledge base from three different years: 2006, 2013 and 2017. The snapshots from 2006 and 2013 were processed by Carvalho et al. (2014) and ready for use. For the 2017 snapshot, we processed it ourselves following the instructions in (Gabrilovich and Markovitch, 2009) and (Carvalho et al., 2014)[5].

7.1 Experiment and Results

Table 2 presents a comparison of the amount of information contained in the three used Wikipedia snapshots. In this experiment, we ran our approach on the two aforementioned datasets but using different concept spaces built from the three different Wikipedia snapshots. The purpose of this experiment is to examine the effect using different versions of the underlying knowledge base has on *C-HTS*.

Table 3 shows the results of the experiment. As we can see, increasing the amount of knowledge in the knowledge base leads, on average, to improvements in hierarchical text segmentation. Although the difference in performance of the three versions is admittedly small, it is consistent across the datasets.

8. Conclusion and Future Work

In this paper, we proposed *C-HTS*, a new Concept-based Hierarchical Text Segmentation approach. The core idea of *C-HTS* is the use of external knowledge to enhance the text representation by adding a semantic layer of concepts that represents the text in a high dimensional semantic space. Relatedness between the atomic units of text is measured using this semantic representation. A Hierarchical Agglomerative Clustering (HAC) algorithm is then used to grow areas of coherent segments. The output of *C-HTS* is a tree-like structure of the input text. We compared *C-HTS* against the state-of-the-art approaches across two different datasets. The results showed that *C-HTS* performed favourably against other approaches.

We also evaluated the influence of the size of the knowledge base that *C-HTS* uses to reason about text. Since *C-HTS* uses Wikipedia as the underlying knowledge base, we measured its performance when using different snapshots of Wikipedia over different years: 2006, 2013 and 2017. The results show that there is a measurable impact upon segmentation performance, and while the difference is small, it is consistent across the two datasets. We also processed a recent Wikipedia snapshot (April 2017) to create a concept space. This processed Wikipedia snapshot along with the implementation of *C-HTS* are publicly available.

Moving forward, and in light of our results, viable future work may involve experimenting *C-HTS* with other familiar languages that have a rich representation in Wikipedia such as French and German.

As text segmentation is widely used as a pre-processing task for Information Retrieval, we plan to use C-HTS with a concept-based retrieval task for content adaptation (Bayomi 2015). The hierarchical structure produced by C-HTS is generated from the use of a concept space that generates new text features automatically. Indexing documents based on their conceptual representation along with these features can be exploited to make the retrieval process more focused.

	2006 Snapshot	*2013 Snapshot*	*2017 Snapshot*
Number of articles	895,000	4,133,000	5,373,241
Concepts used	369,767	1,270,521	1,446,243
Distinct terms	598,391	1,615,525	1,825,353
Concept space size after processing	11 Gb	21 Gb	12.5 Gb[6]

Table 2 Comparison of the three Wikipedia snapshots

	Level	*Moonstone windowDiff*	*Wikipedia windowDiff*
2006 Snapshot	3 (top) 2 (middle) 1 (bottom)	0.347 0.545 0.504	0.365 0.404 0.411
	Average	0.465	0.3933
2013 Snapshot	3 (top) 2 (middle) 1 (bottom)	0.346 0.539 0.509	0.366 0.399 0.405
	Average	0.464	0.390
2017 Snapshot	3 (top) 2 (middle) 1 (bottom)	0.320 0.507 0.488	0.360 0.400 0.409
	Average	**0.438**	**0.3897**

Table 3 *windowDiff* Evaluation of *C-HTS* using different versions of the underlying knowledge source (Wikipedia)

9. Acknowledgements.

The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is cofunded under the European Regional Development Fund.

10. Bibliographical References

Anderka, M. & Stein, B., 2009. The ESA Retrieval Model Revisited. In Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '09. New York, NY, USA: ACM, pp. 670–671.

Bayomi, M., 2015. A Framework to Provide Customized Reuse of Open Corpus Content for Adaptive Systems. In Proceedings of the 26th ACM Conference on Hypertext & Social Media. HT '15. New York, NY, USA: ACM, pp. 315–318.

[5] The technical instructions and snapshots can be found here: http://treo.deri.ie/easyesa/

[6] We indexed the 2017 snapshot in MongoDB v3 that uses the WiredTiger storage engine which applies more compression than the old *mmapv1* engine in MongoDB version used in indexing both 2006 and 2013 snapshots.

Bayomi, M. et al., 2015. OntoSeg: A Novel Approach to Text Segmentation Using Ontological Similarity. In 2015 IEEE International Conference on Data Mining Workshop (ICDMW). pp. 1274–1283.

Bayomi, M. et al., 2016. Towards Evaluating the Impact of Anaphora Resolution on Text Summarisation from a Human Perspective. In the 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, pp. 187–199.

Bayomi, M. & Lawless, S., 2016. ADAPT_TCD: An Ontology-Based Context Aware Approach for Contextual Suggestion. In The Twenty-Fifth Text REtrieval Conference Proceedings (TREC 2016), Contextual Suggestion Track. Gaithersburg, Maryland, USA: National Institute for Standards and Technology, NIST Special Publication pp500-321.

Beeferman, D., Berger, A. & Lafferty, J., 1997. Text Segmentation Using Exponential Models. In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing. pp. 35–46.

Boguraev, B.K. & Neff, M.S., 2000. Discourse segmentation in aid of document summarization. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences.

Davis, Randall, and Douglas B. Lenat. Knowledge-Based Systems in Artificial Intelligence: 2 Case Studies. McGraw-Hill, Inc., 1982.

Budanitsky, A. & Hirst, G., 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics 32.1 (2006): pp13-47.

Carroll, L., 2010. Evaluating Hierarchical Discourse Segmentation. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL Conference. NAACL 2010. pp. 993–1001.

Carvalho, D. et al., 2014. EasyESA: A Low-effort Infrastructure for Explicit Semantic Analysis. In Proceedings of the 2014 International Semantic Web Conference (Posters & Demos), pp. 177-180.

Chang, M.-W. et al., 2008. Importance of Semantic Representation Dataless Classification. In AAAI 2008 vol. 2, pp. 830–835.

Choi, F.Y.Y., 2000. Advances in Domain Independent Linear Text Segmentation. In Proceedings of the 1st North American Chapter of the ACL Conference. NAACL 2000. pp. 26–33.

Choi, F.Y.Y., Wiemer-Hastings, P. & Moore, J., 2001. Latent Semantic Analysis for Text Segmentation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP 2001. pp. 109–117.

Egozi, O., Markovitch, S. & Gabrilovich, E., 2011. Concept-Based Information Retrieval Using Explicit Semantic Analysis. ACM Transactions of Information Systems, p.8:1--8:34.

Eisenstein, J., 2009. Hierarchical Text Segmentation from Multi-Scale Lexical Cohesion. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL Conference, NAACL 2009 , pp.353–361.

Eisenstein, J. & Barzilay, R., 2008. Bayesian Unsupervised Topic Segmentation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP ’08., pp. 334–343.

Gabrilovich, E. & Markovitch, S., 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In IJcAI 2007. Vol. 7, pp. 1606–1611.

Gabrilovich, E. & Markovitch, S., 2009. Wikipedia-based semantic interpretation for natural language processing. Journal of Artificial Intelligence Research, pp.443–498.

Galley, M. et al., 2003. Discourse Segmentation of Multi-party Conversation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL 2003, pp. 562–569.

Givoni, I.E., Chung, C. & Frey, B.J., 2011. Hierarchical Affinity Propagation. Proceedings of the Twenty-Seventh Conference on Uncertainty in AI 2011. pp. 238–246.

Gurevych, I., Müller, C. & Zesch, T., 2007. What to be?-electronic career guidance based on semantic relatedness. In Annual Meeting-Association for Computational Linguistics – ACL 2007, pp. 1032–1039.

Hearst, M.A., 1997. TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. In Journal of Computational Linguistics, 23(1), pp.33–64.

Jain, A.K. & Dubes, R.C., 1988. Algorithms for clustering data, Prentice-Hall, Inc.

Kan, M.-Y., Klavans, J.L. & McKeown, K., 1998. Linear Segmentation and Segment Significance. CoRR.

Kazantseva, A. & Szpakowicz, S., 2014. Hierarchical Topical Segmentation with Affinity Propagation. In COLING 2014. pp. 37–47.

Lawless, S., Lavin, P., Bayomi, M., Cabral, J. P., & Ghorab, M. R 2015. Text Summarization and Speech Synthesis for the Automated Generation of Personalized Audio Presentations. In International Conference on Applications of Natural Language to Information Systems 2015, pp. 307–320.

Malioutov, I. & Barzilay, R., 2006. Minimum Cut Model for Spoken Lecture Segmentation. In Proceedings of the COLING 2006. pp. 25–32.

Morris, J. & Hirst, G., 1991. Lexical Cohesion Computed by Thesaural Relations As an Indicator of the Structure of Text. Computational Linguistics, pp.21–48.

Naili, M., Habacha Chaibi, A. & Hajjami Ben Ghezala, H., 2016. Exogenous approach to improve topic segmentation. International Journal of Intelligent Computing and Cybernetics, pp.165–178.

Navigli, R., 2009. Word Sense Disambiguation: A Survey. ACM Computing Surveys (CSUR),p.10:1--10:69.

Pevzner, L. & Hearst, M.A., 2002. A Critique and Improvement of an Evaluation Metric for Text Segmentation. Computational Linguistics, pp.19–36.

Porter, M.F., 1980. An algorithm for suffix stripping. Program, 14(3), pp.130–137.

Prince, V. & Labadié, A., 2007. Text Segmentation Based on Document Understanding for Information Retrieval. Natural Language Processing and Information Systems, pp. 295–304.

Rafferty, A.N. & Manning, C.D., 2008. Parsing three German treebanks: Lexicalized and unlexicalized baselines. In Proceedings of the Workshop on Parsing German. pp. 40–46.

Reynar, J.C., 1994. An Automatic Method of Finding Topic Boundaries. In Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics. ACL ’94, pp. 331–333.

Stokes, N., Carthy, J. & Smeaton, A.F., 2004. SeLeCT: a lexical cohesion based news story segmentation system. AI communications, 17(1), pp.3–12.

Tellex, S. et al., 2003. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. SIGIR ’03, pp. 41–47.

Utiyama, M. & Isahara, H., 2001. A Statistical Model for Domain-independent Text Segmentation. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, pp. 499–506.

Witten, I. & Milne, D., 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, pp. 25–30.

Yaari, Y., 1997. Segmentation of Expository Texts by Hierarchical Agglomerative Clustering. In Recent Advances in NLP (RANLP’97).

Ye, N. et al., 2008. A Dynamic Programming Model for Text Segmentation Based on Min-Max Similarity. In Information Retrieval Technology - Lecture Notes in Computer Science, pp. 141–152.