Topic segmentation of TV-streams by watershed transform and vectorization

Vincent Claveau, Sébastien Lefèvre

Topic segmentation of TV-streams by watershed transform and vectorization

Vincent Claveau[a], Sébastien Lefèvre[b]

[a]*IRISA – CNRS, Campus de Beaulieu, F-35042 Rennes, France*
vincent.claveau@irisa.fr, tel: +33 (0)2 99 84 74 47, fax: +33 (0)2 99 84 71 71
[b]*IRISA – Univ. Bretagne-Sud, Campus de Tohannic, F-56017 Vannes, France*
sebastien.levevre@irisa.fr, tel: +33 (0)2 97 01 72 66 fax: +33 (0)2 97 01 72 79

Abstract

A fine-grained segmentation of Radio or TV broadcasts is an essential step for most multimedia processing tasks. Applying segmentation algorithms to the speech transcripts seems straightforward. Yet, most of these algorithms are not suited when dealing with short segments or noisy data. In this paper, we present a new segmentation technique inspired from the image analysis field and relying on a new way to compute similarities between candidate segments called Vectorization. Vectorization makes it possible to match text segments that do not share common words; this property is shown to be particularly useful when dealing with transcripts in which transcription errors and short segments makes the segmentation difficult. This new topic segmentation technique is evaluated on two corpora of transcripts from French TV broadcasts on which it largely outperforms other existing approaches from the state-of-the-art.

Keywords: Watershed Transform; Image Segmentation; Vectorization; Topic Segmentation

1. Introduction

Topic segmentation is of high interest in Multimedia information retrieval. Indeed, it is needed to perform automatic structuring of TV streams, a keystone for every processing of such streams, which is still done manually in national archive agencies like the French INA. A way to obtain this structuration is to first transcribe the audio tracks of the TV streams into textual

data, and then perform the topic segmentation from textual data to split the streams into semantic units (e.g., reports).

In this paper we address the problem of topic segmentation of speech in this applicative framework based on a twofold contribution[1]. First, our topic segmentation system is based on the watershed paradigm derived from image segmentation. Second, a key component for this approach is the calculation of the similarity between two successive possible segments; in this paper we present a new technique, called *vectorization* that we recently introduced in the information retrieval field.

The paper is organized as follows. We first present state-of-the-art approaches used for topic segmentation. We then show that topic segmentation and image segmentation have common characteristics (Sec. 3). From this observation we build a topic segmentation method based on the watershed transform, a common morphological tool that identifies segments or regions within a topographic surface. We suggest to build this topographic surface with the help of vectorization which we think is especially suited when dealing with small segments or noisy data such as TV streams (Sec. 4). A first set of experiments, whose goal is to assess the performance of this approach on a standard segmentation benchmark, is presented in Sec. 5. Then, experiments performed on two real TV broadcast corpora are presented and discussed (Sec. 6). Finally, Sec. 7 concludes this work and provides future research directions.

2. Related work

This section is divided into two parts. The first one presents state-of-the-art techniques for topic segmentation. In the second subsection, we compare those techniques and image segmentation ones and show that both fields share many similarities that explain our choice of the watershed transform as a basis for our approach.

2.1. Approaches for topic segmentation

Topic segmentation in TV streams has addressed in several ways in the literature. Multimodal approaches have been proposed, which makes the most of audio, visual and speech features. For instance, segmentation of TV

[1]Preliminary versions and results of this system have been published in Claveau and Lefèvre (2011)

news reports have been explored in TRECVID competition[2] (Amir et al., 2004, for a representative multimodal system). It is worth noting that most of the approaches proposed chiefly rely on training data and do not extend well to other dataset as they focus on superficial clues such as anchor person recognition or background colors. While Poulisse and Moens (2009) have shown the interest of adding multimodal features to improve text-based story segmentation, in the remaining, we only focus on text only approaches.

Various approaches have been applied to speech or text based topic segmentation. Several methods rely on some particularities of the document format, on the detection discourse markers either given by experts (Christensen et al., 2005), or automatically learned (Beeferman et al., 1999). Such techniques require well-formed text and especially a grammatically correct sentence tokenization; they are therefore not suited for texts generated from automatic speech recognition (ASR) systems in which the concept of sentence can rarely match with the oral specifics. Conversely, another kind of approaches is to detect topic changes through document content analysis. These content-based approaches yield high performances and they are less dependent to the document formatting. The overall good quality of modern ASR systems (Ostendorf et al., 2008) makes the use of such approaches on transcribed texts possible (Mulbregta et al., 1999). This is also the approach adopted in our system. In the following, we present representative content-based techniques from the state-of-the-art.

The segmentation process of SEGMENTER (Kan et al., 1998) relies on a representation of the text as weighted lexical chains. Finding the boundaries is thus equivalent to partitioning the resulting graph. The two approaches in DOTPLOTTING (Reynar, 2000) and C99 (Choi, 2000) differ in the way the content is represented, but both rely on the computation of similarities between the candidate segments and then on a clustering based on the resulting similarity matrix. Utiyama and Isahara (2001) propose to use a statistical approach based on hidden Markov models. Here again, the lexical cohesion, key component of the approach, is measured classically with the help of language modeling. The computation of similarity is also at the heart of the TEXT-TILING system (Hearst, 1997), in which a sliding window is used to compare the content before and after each possible boundary. The similarity measure used is inspired from the information retrieval domain

[2] http://www-nlpir.nist.gov/projects/tv2004/tv2004.html

(for instance, a cosine computed from TF or TF-IDF vector representation of the context), and the final boundaries are searched among the places in which the lexical cohesion reaches a significant local minimum. It is worth noting that the approach proposed in this paper can be seen as an modern version of TEXT-TILING in which the similarity computation is done through vectorization (see Sec. 4) and in which the watershed extends the simple boundary detection process used in the original version of TEXT-TILING (cf. next subsection).

These approaches or similar ones have been used on transcribed texts, especially for the story segmentation of broadcast news (Merlino et al., 1997; Stokes et al., 2002; Rosenberg and Hirschberg, 2006; Misra et al., 2010, *inter alia*). To the best of our knowledge, no extensive comparison of them exists for transcribed texts, but all these different approaches have been compared on well-formed texts for different languages (Choi, 2000; Sitbon and Bellot, 2004, respectively on English and French). It is worth noting that all these approaches rely on the word repetition to compute some kind of similarity in order to decide if the topic is changing or not. Therefore, it has been noticed that dealing with segments with very few common words, like short segments or segments with many transcription errors caused by ASR in a noisy environment, is very challenging for this family of approaches. In order to limit the impact of this problem, several authors have proposed to use existing lexical resources or to build them. For instance, Ferret (2009) has compared these two ways, endogenous and exogenous, to bring additional semantic and lexical information to improve the segmentation system. More recently, Guinaudeau et al. (2010) integrated semantically related terms to the segmentation model of Utiyama and Isahara (2001) in order to extend the description of the possible segments. Our approach, thanks to the properties of the vectorization, is expected to be more suited for this kind of problem (cf. Sec. 4).

2.2. Analogies with image segmentation

Although the topic segmentation systems presented above were developed in different theoretical frameworks, it is interesting to highlight some conceptual similarities that they share with our watershed transform framework inherited from the image segmentation domain.

First, the boundaries produced by TEXT-TILING (Hearst, 1997) correspond to areas where lexical cohesion between the text blocks preceding and following the boundaries is associated with a significant local minimum (what

Hearst names *depth score* of valleys). Minima that are selected are associated with areas where cohesion is significantly different from neighboring blocks. No formalization of this boundary detection process is proposed by Hearst (1997), and the choices made are pragmatically justified. Yet, as it is already noted by Hearst (1997, sec 5.3), TEXT-TILING's boundary detection technique tends to miss real boundaries or add spurious ones due to variations along the slopes of the valleys or when dealing with plateaus. On the principles, this approach is very similar to the morphological segmentation using watershed transform on which we build our proposal (see next section), but differs in the implementation. Building on the feedback from the image segmentation literature, it is expected that our watershed formalization yields better results. In particular, our boundary detection technique does not deal with cohesion but its inverse form, considered as a topographic surface and expressed through a gradient, as it is considered as more reliable to extract significant peaks between valleys or plateaus. Moreover, the image analysis field brings us techniques (e.g., considerations on the depth of catchment basins, and subsequent merging strategies) to overcome the weak signal variations that may mislead the boundary detection of TEXT-TILING. Beside that algorithmic difference concerning the boundary detection, our approach can be seen as an improved variant of TEXT-TILING in which the cohesion (or gradient) is computed more cleverly (see Sec. 4).

This comparison with the watershed transform and TEXT-TILING is rather straightforward, but other links between image and text segmentation can also be drawn. This parallel seems, on the one hand, conducive to a better understanding of the topic segmentation methods and how they relate to ours, and on the other hand, a potential source of improvements by making better use of developments in both fields.

For instance, the statistical approach of Utiyama and Isahara (2001), based on hidden Markov models, can be compared to the numerous image segmentation techniques relying on Markov chains or Markov fields (Salzenstein and Collet, 2006). For both domains, these techniques are known to adapt well to noisy data (for example, speech transcripts or textured images), but also requires high computation times.

In the SEGMENTER approach (Kan et al., 1998), the segmentation relies on lexical chains built from weighted links between terms in the text stream. This graph representation, also used in other segmentation systems, is thus similar to the very popular graph-based image analysis framework (Shi and Malik, 2000). Depending on the size and graph morphology, these approaches

also tend to suffer from a high algorithmic complexity.

The DOTPLOTTING (Reynar, 2000) and C99 (Choi, 2000) systems, although based on different representations of the text, both rely on a clustering step to group coherent segments. This clustering step is also used in many image segmentation systems (Gonzalez and Woods, 2008)

3. Topic segmentation as morphological segmentation

In this section, our topic segmentation approach is presented. By making an analogy with the image segmentation problem (subsection 3.1), we show that the text segmentation problem can also be modeled so as to be solved with a watershed transform (subsection 3.2), extending then the seminal TEXT-TILING approach of Hearst (1997). As a key component, the way to compute the similarity between parts of the stream is discussed in subsection 3.3.

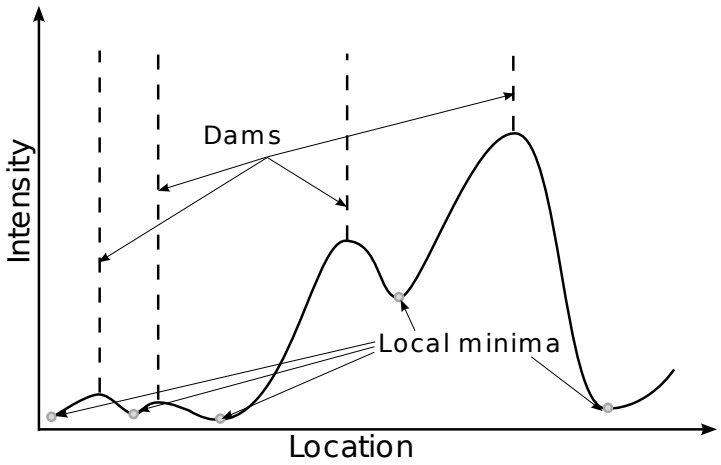
3.1. Morphological segmentation

Mathematical morphology is both a rich theoretical framework and a complete toolbox mostly used in the image processing community. Particularly, it has been extensively used for image segmentation, which aims at splitting an input image into a set of uniform regions given a predefined uniformity criterion (intensity or colour, texture, etc.). The most famous morphological method for image segmentation is certainly the watershed transform.

We recall very briefly the principle of watershed-based segmentation (Vincent and Soille, 1991, for a comprehensive presentation). The image I to be segmented is first represented as a function f describing a topographic surface. Watershed lines identified on this surface are then associated to region frontiers resulting from the segmentation process. One common way to implement it, called the immersion paradigm, is to simulate the progressive flooding of the surface starting from its local minima, and then to build dams to avoid merging water from two different catchment basins. At the end of the process, dams correspond to the watershed lines or, in other words, to the region frontiers (see Fig. 1).

Most often, this approach is not directly applied on the image I to be segmented. Before applying the segmentation, an image transform is rather performed as a preprocessing in order to highlight values of edge pixels and to lower pixel values in homogeneous areas. A gradient (noted ∇ hereafter)

Figure 1: Example of a watershed in 1-D: the altitude of each pixel p is defined by its intensity $I(p)$ in the image to segment.



is thus usually computed to enhance transition areas (which generally correspond to object frontiers). Therefore, the function f on which the watershed transform is applied is usually already the result of a transform of the initial image. In practice, various gradient computation methods can be used. Its choice is of high importance, since it will directly influence the segmentation result produced by the watershed method.

As noted by Vincent and Soille (1991), although the watershed concept is simple, its formalization is more complex and may take many forms. We reuse here the notations from Roerdink and Meijster (2001). The watershed relies on the notion of topographical distance; for a continuous function f defined over a domain D, the topographical distance between points p and q of D is defined as

$$T_f(p,q) = \inf_{\gamma} \int_{\gamma} \|\nabla f(\gamma(s))\| ds \tag{1}$$

that is, the infimum over all paths (smooth curves) γ inside D with $\gamma(0) = p$ and $\gamma(1) = q$. The path of steepest slope between p and q is the one with the shortest T_f-distance.

Let us now consider the minima of f, noted $\{m_k\}_{k \in I}$. The catchment basin $CB(m_i)$ of a minimum m_i is defined as the set of points $x \in D$ which are topographically closer to m_i than any other minimum m_j, that is:

$$CB(m_i) = \{x \in D | \forall j \in I \setminus i, f(m_i) + T_f(x, m_i) < f(m_j) + T_f(x, m_j)\} \tag{2}$$

Finally, the watersheds of f are defined by the set of points which do not belong to any catchment basin:

$$WS(f) = D \setminus (\cup_{i \in I} CB(m_i)) \tag{3}$$

In the case of digital images, this general definition has given birth to many implementations to deal with discrete data (i.e. pixels are defined as positive integers taking values in $\mathbb{N}^d$ or most often as subset of it, e.g., $[0, 255]$ for greyscale images). In this paper, we adopt the algorithmic approach by immersion proposed by Vincent and Soille (1991), which relies on a recursive definition of the building of the basins (see below for the adaptation of this algorithm in our case). More approaches as well as other definitions and an in-depth discussion of their respective advantages and drawbacks are discussed in (Roerdink and Meijster, 2001).

3.2. From image to text

The analogy between image and text segmentation can be drawn very simply. The pixel is the base element in the image and is described by its greylevel or color/multispectral values. Its equivalent in texts is the sentence (or sometimes the paragraph) which is described by the words it contains. In our framework of multimedia information retrieval, our texts are obtained from automatic transcription. Thus, the transcribed utterances are the minimal units of the text (i.e., they are equivalent to image pixels) and topic breaks will be sought between them. In both case (pixels and texts), elementary units take their coordinates in a discrete space (grid or line).

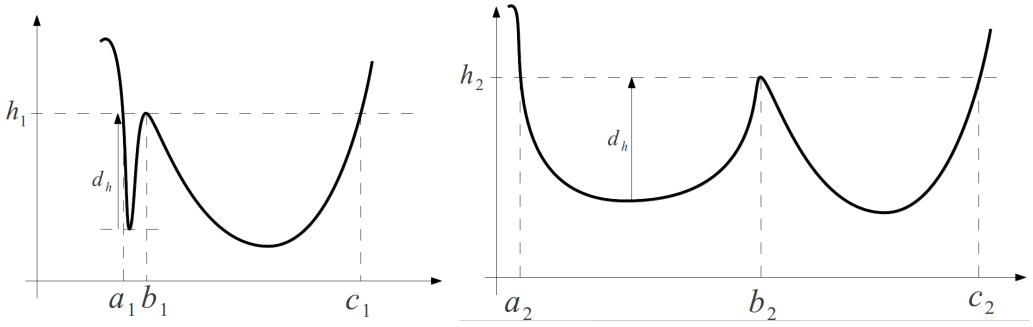
Besides, our texts are flows of utterances. They are then represented as 1-D signals, while images are most often 2- or 3-dimensional. However, nothing prevents the watershed technique to be applied on a single dimension as shown in Fig. 1. Thus our approach relies on a gradient computed on the sequence of utterances, and topic breaks are identified using the watershed transform (Claveau and Lefèvre, 2011). Gradient computation, which is a key step of the segmentation process, is detailled in Sec. 4. The watershed technique used here follows the immersion paradigm described previously, but simplified as we handle a 1-D signal. While the utterance indices are discrete, their associates values computed with the vectorization process are here continuous.

More precisely, let $f : D \rightarrow \mathbb{R}$ be this signal, defined on the stream of utterances D, with h_{min} and h_{max} the minimum and maximum values

observed in f. In this case, the immersion algorithm of Vincent and Soille (1991) is a simple recursion on the "altitude" h increasing from h_{min} to h_{max}. The basins associated with the minima of f are successively expanded. Let X_h denote the union of the set of basins computed at level h. Any point of D at level $h' \succ h$ (with $\succ$ defining the successive operator, i.e. $h' \succ h \Leftrightarrow \nexists k,\ h < k < h'$) can be either 1) a new minimum, or 2) an extension of a basin in X_h if p is adjacent (previous or next utterance) to one and only one point in a basin of X_h, or 3) a watershed point if it is adjacent to two different basins of X_h. In the first case, a new basin is added and in the second case, the point is included in the existing basin, resulting in an updated set $X_{h'}$. Finally, the watershed of f is the complement of $X_{h_{max}}$ in D, that is, the points labeled as watershed.

A known caveat of the immersion algorithm is that it tends to produce over-segmentation. In order to prevent this, two classical strategies are used. First, as a preprocessing, we have included a gradient smoothing step to remove irrelevant local minima. It is simply done by taking at each point the median value of the function over the three preceding and following utterances. TEXT-TILING algorithm proceeds similarly by applying a sequence of mean filters on the signal. As a post-processing of the immersion algorithm, a basin merging is performed following a strategy inspired by Najman and Schmitt (1994). It is based on the dynamics and volumes of the basins; the dynamic of a basin is defined by the minimum height which has to be overcome in order to reach a basin with lower or equal minimum altitude. Merging between the two basins occurs if the volume of water that is caught in the first basin up to the highest in-between altitude is significantly lower than the one in the second basin. This is illustrated on Fig. 2: on the left, the two basins can be merged (i.e. the dam in b_2 is removed), since the volume of the left basin, defined as $\int_{a_1}^{b_1} h_1 * f(x)dx$, is small compared with the right basin's one, i.e. $\int_{b_1}^{c_1} h_1 * f(x)dx$. To the contrary, on the right, although the right basin and dynamic of the left basin are identical to previous case, the two basins are not merged. It is worth noting that TEXT-TILING, given the same signal, would handle the two cases in a same way (i.e. the minima of the first basins in the two configurations would both be kept as boundaries or both rejected depending on global characteristics of the whole signal). This merging process is repeated until no more merging is proposed. In the experiments proposed in this paper, the dynamics and volume thresholds are simply the following ones: the merging of basin X with basin Y is allowed if

Figure 2: Merging based on the dynamic and volume of basins. On the left, merging is allowed; on the right the basins are not merged.



the dynamics and volume of X are (both) lower than Y's ones (see Sec. 6.4 for a discussion on this setting).

3.3. Text similarity as topic gradient

A gradient is computed between each utterance. In other words, we compute the similarity using the vectorization principle between previous and next utterances. Let us note that we do not compare only the previous to the next utterance, but we also consider the n previous ones vs. the n next ones (similarly to common approaches for topic segmentation such as TEXT-TILING). Computing similarities or distances between texts is a common task in Natural Language Processing and Information Retrieval. One of the best known technique is the TF-IDF/cosine (Salton, 1975): the texts, considered as bags-of-words, are represented in a vector space; each dimension represents the importance of a word in the text, given by its TF-IDF weight (TF and IDF respectively stand for Term Frequency and Inverse Document Frequency). More formally, the weight w of a term t in the text d is defined by:

$$w_{TF-IDF}(t,d) = tf(t,d) * \log(N/df(t)) \tag{4}$$

where tf is the number of occurrences or frequency of term t in the considered text, df is its document frequency, that is, the number of texts in which it appears, N is the total number of texts. The complete vector is thus: TF-IDF$(d) = [w_{TF-IDF}(t_1,d), w_{TF-IDF}(t_2,d), ..., w_{TF-IDF}(t_n,d)]$, where $t_1, \ldots, t_n$ are the words occurring in the whole text stream to segment. Once represented as (TF-IDF weighted and possibly normalized)

vectors, two texts can then be compared by computing a cosine similarity or an L2 distance between them. Note that for normalized vectors, cosine and L2 produce comparable results since for any vectors u and v: $\delta_{L2}(u,v) = \sqrt{2 - 2 * \delta_{cos}(u,v)}$.

In our experiments (Sec. 5 and 6), a more modern weighting scheme is also tested. This similarity measure, called Okapi-BM25 (Robertson et al., 1998), can be viewed as an improved TF-IDF while yielding usually much better results. Due to its good results in various IR experiments, this weighting scheme is often considered as a challenging baseline. Its definition is given by

$$\begin{aligned} w_{BM25}(t,d) &= TF_{BM25}(t,d) * IDF_{BM25}(t) \\ &= \frac{tf(t,d) * (k_1+1)}{tf(t,d) + k_1 * (1 - b + b * dl(d)/dl_{avg})} * \log \frac{N - df(t) + 0.5}{df(t) + 0.5} \end{aligned} , \tag{5}$$

where $k_1 = 2$ and $b = 0.75$ are constants, dl is the text length, dl_{avg} is the average text length.

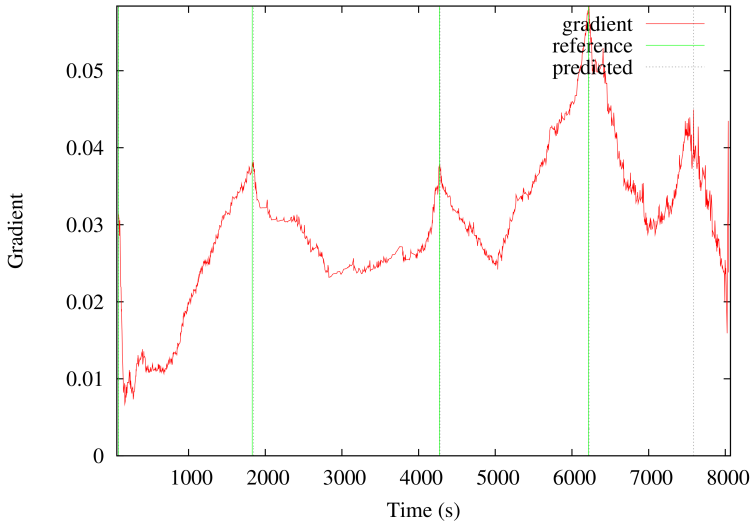
As it was announced in the introduction, a new similarity measure called vectorization is also proposed in this paper. Its description is given in the following section.

Whatever the similarity measure used and similarly to some image gradient computation methods, we give more importance to close utterances and less importance to utterances which are far from the candidate edge. This is ensured through a simple convolution with a kernel (e.g., Gaussian kernel). Let us notice that the way the convolution is applied depends on the way the documents are represented and on the similarity/distance model chosen. With a TF-IDF or Okapi vector model, this convolution is very simply implemented: when computing $tf(t,d)$, the occurrence of a word counts for one in the breath group which is the closest from the candidate edge, but counts for less when considering an occurrence from a breath group further of the candidate edge. In our experiments, a linear penalty is applied. From now, we write $\mathcal{C}_{prev}(i)$ (respectively $\mathcal{C}_{next}(i)$) the result of the convolution operator applied on utterance i and those which are preceding (respectively following) it. In the experiments reported in this article, the default size of the context is 100 breath groups or sentences; see Sec. 6.4 for experiments and discussion on this point. Formally, the gradient, computed with TF-IDF weights for instance, is defined by

$$\nabla(i) = \delta_{L2}(\text{TF-IDF}(\mathcal{C}_{prev}(i-1)), \text{TF-IDF}(\mathcal{C}_{next}(i))) \tag{6}$$

In the experiments reported below, utterances are represented by their starting time. For a given time index, the higher the gradient is, the more important the dissimilarity between previous and next groups is. In other words, significant local maxima of gradient values indicate a topic break. Fig. 3 shows an example of gradient computed with vectorization on one document of one of our experimental collection (see below). This document contains 4 segments whose boundaries are indicated in plain green; the topic limits detected by our approach are represented by dashed lines. We recall that in our approach, we do not apply the watershed transform directly on the input signal (in red) but rather filter it with some post-processing techniques to remove local variabilities and thus ease the detection of the segment boundaries. We then expect that topic segment boundaries correspond to the gradient most significant maxima extracted by the watershed transform.

Figure 3: Gradient vs. starting time of utterances



4. Vectorization as a robust gradient

4.1. Vectorization principles

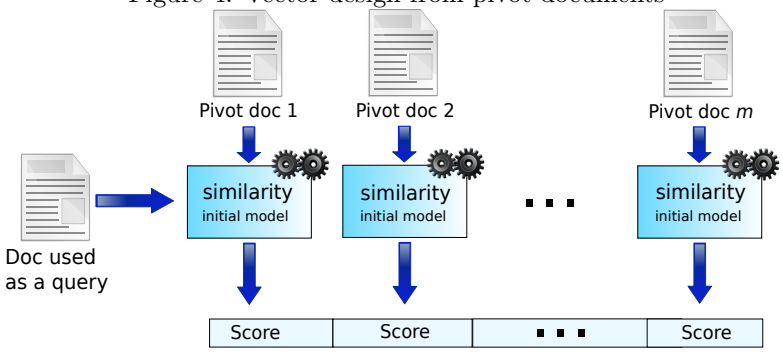
Vectorization is an embedding technique which aims to project any similarity computation between two documents (or one document and one re-

quest in the context of IR) in a vector space. It has been introduced and experimented in a standard IR scenario (Claveau et al., 2010) where it has shown to provide both a low complexity and accurate results. It can also be linked to previous work that has been made in the image segmentation field (Derivaux et al., 2010) where the watershed technique operates on a gradient image built from fuzzy classification of pixel values. In the remaining of the section, we recall vectorization main characteristics.

Its principle is relatively simple. For each document of the considered collection, it consists in computing with an initial similarity measure (e.g., standard similarity measure used in IR), whatever it is, some proximity scores with m pivot-documents. These m scores are then gathered into a m-dimensional vector representing the document (cf. Fig. 4).

Comparing two documents (or a document and a request) can then be performed in a very standard way in this vector space (e.g., using an L2 distance). Many algorithms are available to compute or approximate very efficiently such distances.

Figure 4: Vector design from pivot-documents



More formally, we note $\text{Vect}(D, \mathcal{P}, Sim)$ the vector representing the document D built from the initial similarity measure Sim on pivot-documents $\mathcal{P}$. For instance, $\text{Vect}(D, [P_1, P_2, P_3], \text{TF.IDF/cosine})$ is a 3-dimensional vector; its first component is the similarity score between the document D and the pivot-document P_1 returned by a system using TF.IDF representation associated to the cosine distance measure (as explained in Sec. 3.3, it corresponds to a very standard way to compute similarities in the IR field), and so on for the next components.

4.2. Properties

Several studies have suggested, for various reasons, representations different from standard vector model. Among these, the *Generalized Vector Space Model* (GVSM) (Carbonell et al., 1997), responds to criticism that the words are a poor basis for the vector space as they are not independent of each other. The GVSM uses the dual space, where the documents form the basis of this space, and can be more easily considered as independent.

The many LSI (Latent Semantic Indexing) variants (Deerwester et al., 1990) also belong to this family. This includes techniques using pLSA (probabilistic Latent Semantic Analysis) (Hofmann, 1999), principal component analysis (PCA) (Berry and Martin, 2005), LDA (Latent Dirichlet Allocation) (Blei et al., 2003), and even random linear transforms (Vempala, 2004). These methods all start from the classic vector model which they transform the matrix terms $\times$ documents, noted $\mathcal{M}$ hereafter, with the primary effect of reducing it. Our approach, when used on a vector system, shares many links with these techniques. But it is more generic because it applies to any form of IR model, provided that it outputs a score to represent the relevance of a document to a query. For instance, in the LSI model, a Singular Value Decomposition is applied to the term-document matrix $\mathcal{M}$ (possibly TF-IDF weighted), resulting in $\mathcal{M} = USV^T$, where U is a term–latent topics matrix, S is the diagonal matrix containing the singular values and V is a document–latent topic matrix. The projection of the documents in the new space defined by the topics is $U^T\mathcal{M} = SV^T$. In our approach, it is possible to mimic this with a particular setting: we adopt a vector representation for the documents, resulting in the same (possibly TF-IDF weighted) matrix $\mathcal{M}$, the pivot-documents are crafted such that each pivot contains the word representation of a topic (that is, $\mathcal{P} = U$), we choose the cosine as the similarity function; with documents and pivots being normalized, it is equivalent to a scalar product. In that case, the document D, here considered as a vector, is represented by $\text{Vect}(D, \mathcal{P}, \text{TF.IDF/cosine}) = \mathcal{P}^T D$, and more generally, the whole document collection is represented in the new vector space by $\mathcal{P}^T\mathcal{M}$, exactly as for LSI. Of course, LSI provides a way to build the orthogonal base U, while vectorization does not for $\mathcal{P}$, but as it is suggested by Random indexing results, this property is not necessary for achieving good results. In our task of story segmentation, building $\mathcal{P}$ from random segments of the text stream seems to ensure a good representation of the different topics.

It is also important to notice that vectorization results in a change of representation space, contrary to existing works consisting rather of a dimension

reduction or a distance approximation (e.g., Abraham et al., 2006). This space transform offers several nice properties which will be discussed here.

The first interest of this embedding is to reduce complexity when the initial similarity computation may be computationally expensive (e.g., some graph comparison computations used in complex IR systems). In an IR context, vectors associated with each document may be built offline, and when a request has to be processed, we only need to compute its similarity with the m pivot-documents rather than with all documents in the collection. This property is nevertheless not useful in the context of a segmentation task.

The second nice property comes from the fact that two documents will be considered as similar if they are similar to the same pivot-documents. This indirect comparison, or second-order affinity, let us compare two textual documents which do not share any common word. This property will be helpful in our segmentation task. Indeed, it will solve the problem brought by the lack of repetition between utterances. This problem is particularly noticeable when the segments to be compared are of short duration (i.e., they will contain less words, and thus will share only a few words in common in the best case, and no common word in the worst case).

4.3. About complexity

Techniques for rapid calculation of distances in vector spaces have also been studied. These techniques can allow our approach to dramatically reduce its complexity. To save processing time, these techniques address either the completeness of the search or the accuracy of distance calculation. Indeed, the *hashing*-based techniques, used for retrieving similar documents or detecting plagiarism (Stein, 2007) tackle the completeness: the space is divided into portions, and the research is conducted on a subset of these portions. In this family, the LSH (*locally-sensitive-hashing*) approach (Datar et al., 2004) uses hash functions to restrict the search space to an hyper-ball centered on the approximate query point with a radius set by the user. An exact L2 distance is then calculated with every element of this ball.

The NV-tree (Lejsek et al., 2008) pushes this approach further: it builds portions from the concatenation of multiple random projections of points in space, analyze one portion for each query, which portion size is calculated to generate a single disk access. In addition, the NV-tree computes approximate L2 distances, which further reduces the cost of the matches. Finally, it provides results in $\mathcal{O}(1)$ (constant time corresponding to a single disk access), whatever the number of points in the space.

4.4. Vectorization for segmentation

In experiments described in the following section, the initial similarity measure used in the vectorization process is an L2 distance associated with a weighting of utterances by $\sqrt{TF}$. It means that we first represent each breath group by a sparse vector in which each dimension represents a word; the value for this dimension is the square root of the number of occurrences of the word in the breath group. The same is done for the pivot-document. The distance between the breath group vector and the pivot vector is computed with an L2 distance; the resulting value forms one of the dimensions of the new vector.

As explained in the previous section, in our segmentation system, we give a greater importance to utterances close to the candidate edge (and lower importance to distant utterances) with a simple convolution whose results are noted $\mathcal{C}_{prev}(i)$ and $\mathcal{C}_{next}(i)$. With these notations, the gradient computed with vectorization is thus formally defined by:

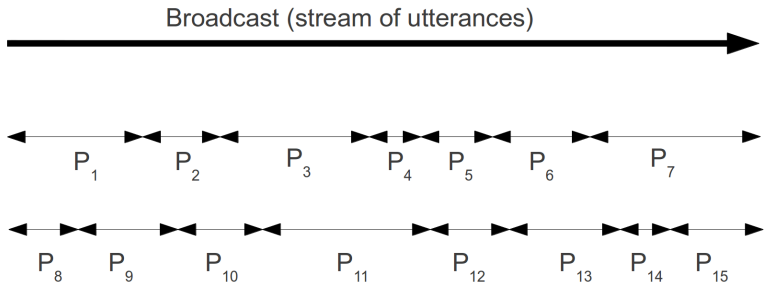
$$\nabla(i) = \delta_{L2}(\text{Vect}(\mathcal{C}_{prev}(i-1), \mathcal{P}, \sqrt{TF}/L2), \text{Vect}(\mathcal{C}_{next}(i), \mathcal{P}, \sqrt{TF}/L2)) \quad (7)$$

In the experiments reported below, the pivot-documents that we are using are sequences of sentences (respectively, utterances when dealing with transcribed texts) built from random splits of the considered document (resp., broadcast). More precisely, several random splits are generated providing overlapping segments of different sizes as illustrated in Fig. 5. The only parameter given is an approximate number of pivots expected; the size of each pivot is randomly selected but cannot be lower than two sentences (to prevent too specific pivots) or greater than half the document (to prevent too broad and uninformative pivots). It is obvious that the number of pivots must be greater than the number of segments to ensure good results. But it worth noting that the number of pivots, if above this minimal threshold, does not impact the results: indeed, additional pivots add redundancy in the representation which is not harmful to the relevance of the distance computation (however, it adds computational complexity). In the experiments presented below, we have set this number to 5 000, that is expected to be far greater than the number of segments.

5. Preliminary experiments

In this section, we provide a first comparison between different variants of our segmentation technique and state-of-the-art systems on well-formed

Figure 5: Pivot-documents randomly generated from the document to be segmented



written texts. To do so, we use the benchmark developed by Sitbon and Bellot (2004) that we present hereafter before introducing the evaluation measures and discussing the obtained results.

5.1. Experimental data

As it was previously mentioned, the different existing segmentation algorithms have been tested and compared with specially crafted evaluation data. Such test sets have been developed for many languages, including English (Choi, 2000) or French (Sitbon and Bellot, 2004), and are usually artificially generated by concatenating segments from different sources into one text stream.

For these preliminary experiments, we use the test set developed by Sitbon and Bellot (2004). This test set is in French (as the TV transcripts used in our main experiments) and has already been used to compare standard segmentation algorithms (after their adaptation to French)[3]. It is composed of different subsets obtained by concatenating parts of articles of the newspaper *Le Monde*. The articles are chosen within a same category (sports, arts, politics...) and are of variable lengths. Another test subset is built the same way by concatenating verses of the Bible.

5.2. Evaluation

Different scores have been proposed to evaluate the quality of segmentation systems. Beeferman et al. (1999) have shown that computing Recall and

[3]We thank L. Sitbon and P. Bellot for making this test set available for our experiments as well as the French versions of the state-of-the-art segmentation algorithms.

Precision without any preprocessing may lead to inconsistencies and propose the Pk-score, which has been widely used. Yet, Pevzner and Hearst (2002) advocate that Pk-score, even if it is better than Recall and Precision, presents some failures, especially for the following conditions:

- missing boundaries are more penalized than false alarms;
- near-miss errors are heavily penalized compared with false alarms and missing boundaries;
- when a boundary is added implying new segments of size smaller than k, it is not detected and thus not added to the score;
- the fixed-length window on which the score is based is not suited for great variations of the segment sizes;
- the meaning of the score is not clear since it cannot be interpreted as an error percentage as it may seem.

Based on that, Pevzner and Hearst (2002) have proposed a variant of the Pk-score called WindowDiff, which is usually preferred for evaluating segmentation systems, and can be seen as an error rate. Thus, lower WindowDiff scores indicate better segmentation accuracy. It is defined as:

$$WD(ref, hyp) = \frac{1}{N-k} \sum_{i} |b(\mathrm{ref}_i, \mathrm{ref}_{i+k}) - b(\mathrm{hyp}_i, \mathrm{hyp}_{i+k})| > 0 \quad (8)$$

where $b(x_i, x_j)$ is the number of boundaries between i^{th} and j^{th} sentences (or any other minimal units, depending on the segmentation task considered) in the stream x, which contains N sentences. Different k values can be set, but it is standard to define it as:

$$k = \frac{N}{2 * \text{number of segments}} \quad (9)$$

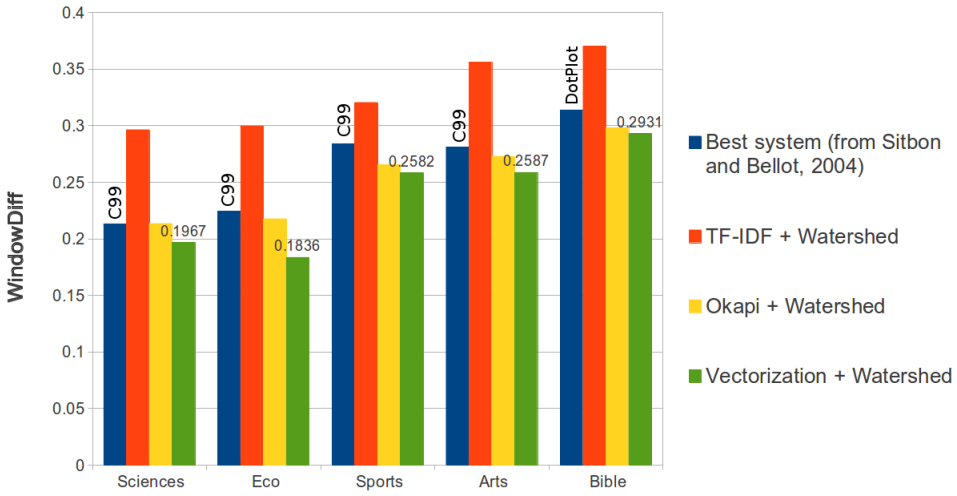
This is the definition that we adopt in the experiments reported below.

5.3. Results

Fig. 6 presents the results obtained by our segmentation system with several variants in the way the similarity between potential segments is computed: TF-IDF, Okapi and Vectorization. For comparison purposes, several

state-of-the art systems were tested on this dataset: DotPlot (Reynar, 2000), text-tiling (Hearst, 1997) and C99 (Choi, 2000) algorithms, as implemented by Choi (2000) and adapted to French by Sitbon and Bellot (2004). We indicate in the table the best-performing system for each test subset along with its score. Since most of these state-of-the-art systems rely on different parameters, the results reported here are the best ones obtained with the optimal parameter settings found.

Figure 6: Performance (measured with the WindowDiff error rate) of segmentation systems on the test set of Sitbon and Bellot (2004)



These results clearly shed light on the importance of the similarity function. Indeed, a TF-IDF based system yields results significantly lower than the state-of-art's ones, but its more modern variant, Okapi, allows us to obtain a WindowDiff score slightly better than these ones. Last, computing similarities with our vectorization approach makes it possible to gain further accuracy. These experiments on this artificial dataset with clean, well-formed texts, validate the interest of the watershed approach, as well as the similarity computation through Vectorization.

6. Experiments

Based on the good results yielded by our system on the written dataset, we examine in this section its use on a real-life application using speech tran-

scripts. These data are described in the next subsection and the experimental setting and results are then presented.

6.1. Experimental data

Our experiments are performed on two French TV broadcast corpora for which the topic segmentation is of high interest. The first corpus is a set of 60 TV news of the France 2 channel (called *News* further). Each of these sample has been broadcasted in the beginning of 2007 and is 40 minutes long. The second corpus is made from TV reports: 12 samples of *Envoyé spécial* (2008, 2 hours long each), and 16 *Sept à huit* (2008, 1 hour long each). This corpus is called *Reports* in the following experiments.

These corpora (Guinaudeau et al., 2010) have different properties in terms of number and duration of topic segments. Thus, it allows us to evaluate robustness of topic segmentation methods. The *News* corpus contains 1180 segments while the *Reports* corpus only contains 140 segments.

The reference segmentation (i.e., ground truth) has been independently built by a user who was not involved in the design of a topic segmentation system. Since there is no consensus on the topic definition in the IR or NLP fields, it has been considered here that a topic change occurs for each report change. Despite this assumption being not always valid (in particular in the *News* corpus in which several successive reports may be considered as related to the same topic), it is relevant since it corresponds to an actual and well-defined applicative need.

Audio tracks of these two corpora have been automatically transcribed using the speech recognition system IRENE (Huet et al., 2010). This system has been initially designed for transcribing radio broadcasts, including news, and is thus well-suited for our corpora. For these data, its Word Error Rate is about 20%, but this rate highly varies among the documents (e.g., anchor person speech vs. noisy outdoor speech). Transcriptions are finally part-of-speech tagged using TreeTagger[4], and only names, verbs, and adjectives are kept and stemmed.

6.2. Evaluation and comparison

For the evaluation on these new data, we use the same measure than in the previous section. Yet, in order to compare our results with existing ones,

[4] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger`

we also implement Recall (R), precision (P), and F1-score (F1) to evaluate the quality of the methods tested. These measures were preferred by Guinaudeau et al. (2010); in order to prevent the biases reported in the literature (Beeferman et al., 1999), an alignment between the tested segmentation and the reference one is first performed before computing R, P and F1. Also, as stated by Guinaudeau et al. (2010), a segment frontier is considered as correct when it is located at less than 10 seconds of a reference frontier. This flexibility is needed for these transcription-based techniques to compensate the difference between the end of breath groups and the actual end of segments.

In order to show the relevance of our contribution, we compare the results obtained by our method to those produced by a baseline and by several existing systems on the same corpora. The baseline simply consists in dividing the document into as many (equal length) segments as there are segments in the reference. Concerning the existing systems, we use the DotPlot (Reynar, 2000), text-tiling (Hearst, 1997) and C99 (Choi, 2000) algorithms, as implemented by Choi (2000) and adapted to French by Sitbon and Bellot (2004). We also report the results, when available, of the system of Utiyama and Isahara (2001) (as implemented by Guinaudeau et al. (2010)) and the best results obtained from the system of Guinaudeau et al. (2010). Moreover, in order to assess the impact of vectorization similarity measure, we also provide results obtained by our watershed approach using instead standard distances, e.g., TF-IDF/L2 and Okapi-BM25. For a fair comparison, it is worth noting that DotPlot, C99 and the baseline take as input the number of expected segments, while the other approaches do not.

6.3. Results

As it is noted by Huet et al. (2008), the data on which we focus, transcribed TV broadcasts, have different characteristics that are detrimental to complex processing such as our topic segmentation task. First, some difficulties are related to the very nature of our data and our task. Indeed, in the TV collections that we manipulate, the topic segments may be very short. Moreover, they contain few repeated words, due to the journalistic style which voluntarily prefers synonyms, paraphrases or pronominal references in order to avoid repetition. In the corpus used for our experiments, Guinaudeau et al. (2010) report that a word occurs 1.8 times on average in a topically coherent segment in the news broadcast and 2.0 times in the reports on current affairs. As it has been said in Sec. 2, this problem of

lack of repetition is usually tackled with the addition of lexical resources. These experiments are expected to highlight the interest of our vectorization process as a simple and integrated way to counter this problem.

Tables 1 and 2 show results obtained by all the systems on the two corpora. In both cases, we can observe that our system (Vectorization + Watershed) yields better results than existing systems, whatever the evaluation measure considered. It is also interesting to note that the watershed approach performs well, even combined with a simple similarity measure such as TF-IDF/cosine. As expected, the superiority of Vectorization as a similarity measure is particularly observable on the *News* corpus, since this corpus contains very short segments, thus making the direct computation of the gradient as done in TF-IDF + Watershed approach unreliable. Moreover, in order to better understand the interest of using Watershed for topic segmentation, it is interesting to compare more deeply the approach introduced in this paper and TEXT-TILING. Indeed, the TEXT-TILING approach aims at finding topic breaks where lexical coherence between previous and next text blocks is linked to a significant local minimum. As explained previously, this seminal approach can be seen as a particular case of ours, but the similarity is computed based on a TF or TF-IDF/cosine measure, and the minima identified as those below a threshold based on the mean similarity. In order to identify the influence of each component, we also use our vectorization score within TEXT-TILING, that is, we use the TEXT-TILING boundary detection process (implementation of Sitbon and Bellot (2004)), instead of the watershed transform. Since TEXT-TILING expects a cohesion score, the score actually used is $f(x) = max_i(\nabla(i)) - \nabla(x)$. But as it appears in both experiments, the results are far below the ones of the watershed, and even below the original score of TEXT-TILING on the News corpus. A close examination of the results seems to indicate that TEXT-TILING fails due to a wrong strategy for detecting the number of boundaries to be kept. It is based on the average and standard deviation of the signal, and is not suited to the form of the vectorization signal, which shows large variations, thus resulting in keeping too few boundaries.

6.4. Segmentation parameters

As any segmentation algorithm, our approach relies on different choices and parameters. In the general case, the segmentation is to be used in an unsupervised way, that is, with no manually-segmented training data that would help to optimize these parameters. Thus, it is interesting to have a

Table 1: Performance of topic segmentation systems on *News* corpus

Methods	P	R	F1	WD
Baseline	15.39	13.39	15.39	0.546
Utiyama and Isahara (2001)	57.6	61.4	59.44	-
DotPlot (Reynar, 2000)	36.42	36.42	36.42	0.4472
c99 (Choi, 2000)	50.25	50.25	50.25	0.3646
text-tiling (Hearst, 1997)	47.25	35.96	38.73	0.313
text-tiling + Vectorization	48.6	29.47	36.69	0.351
Watershed + TF-IDF	48.17	49.82	49.40	0.3421
Watershed + Okapi	64.06	56.49	60.04	0.2571
Watershed + Vectorization	**72.44**	**66**	**69.07**	**0.2269**

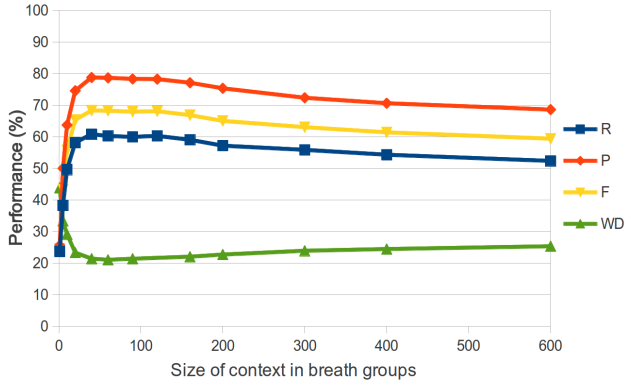
Table 2: Performance of topic segmentation systems on *Reports* corpus

Methods	P	R	F1	WD
Baseline	1.9	1.9	1.9	0.364
Utiyama and Isahara (2001)	75.3	**73.6**	74.4	-
DotPlot (Reynar, 2000)	49.49	49.49	49.49	0.2125
c99 (Choi, 2000)	57.42	57.42	57.42	0.1893
text-tiling (Hearst, 1997)	25.96	21.27	23.38	0.3456
text-tiling + Vectorization	46.44	22.2	30.03	0.2611
Watershed + TF-IDF	59.32	60.93	60.12	0.1844
Watershed + Okapi	72.91	65.89	69.22	0.1428
Watershed + Vectorization	**77.98**	72.57	**75.18**	**0.1181**

closer look to some parameters to compare different settings to the default ones. In Fig. 7, we examine the influence of the size of the context (number of breath groups on the X axis) taken into account when computing $\mathcal{C}_{prev}(i-1)$ and $\mathcal{C}_{next}(i)$. It appears that as soon as a minimum size is considered (about 15), this parameter has little impact on the different evaluation measures considered. The default value chosen (100) gives almost optimal results. The same behavior with the same values is also observed on the other collections (experiments not reported here), which tends to show that this parameter is not critical and does not need fine tuning.

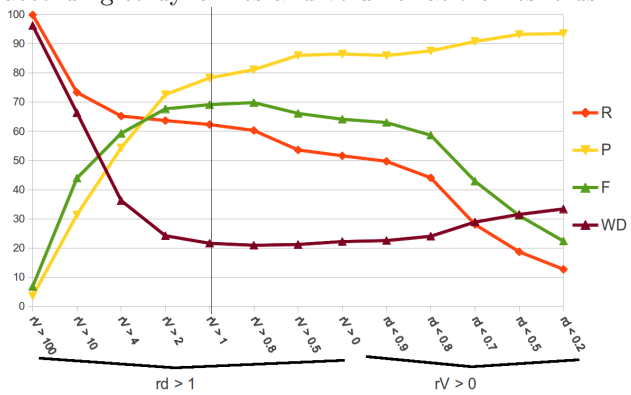
As explained in Sec. 3.2, the immersion algorithm implementing our watershed tends to produce over-segmentation. The merging strategy that we use to prevent it is inspired by Najman and Schmitt (1994) and relies on the

Figure 7: Effect of context size on the News corpus: Precision, Recall, F-score and WindowDiff according to number of breath group used to compute $\mathcal{C}_{prev}(i-1)$ and $\mathcal{C}_{next}(i)$.



dynamics (which is related to the depth of basins) and volume of the basins to decide whether or not two adjacent basins should be merged. In order to study the importance of this merging step, we evaluate the results obtained with different settings for these two constraints. In Fig. 8, we progressively relax the volume and then the dynamics constraints: the X axis indicates the minimum ratio of volumes (rV is the ratio of volume of a basin B over the volume of basin A when considering merging A into B) and then dynamics (rd) to allow a merging. For instance, the first setting on the left on the X axis allows a merging of a basin A into a basin B if the dynamics of B at least higher than A's one and the volume of B is 100 times higher than A's one. Several observations can me made. First, the leftmost configuration

Figure 8: Effect of merging parameters on the News corpus: Precision, Recall, F-score and WindowDiff according to dynamics and volume ratio thresholds.



actually corresponds to no merging at all, and thus provides many irrelevant boundaries. On the other side, the rightmost configuration corresponds to many mergings, which results in long segments and too few boundaries. It is also interesting to note that several configurations yields good results; the default thresholds used in our experiments, symbolized by the vertical line, are among these optimal configurations. The best setting both in WindowDiff and F-score is reached at $rd > 1$ and $rV > 0.8$ (that is, it allows some more mergings than our default strategy). Last, comparing our default configuration to the $rV > 0$ configuration illustrates the interest of adding the volume constraints to Najman and Schmitt (1994)'s approach only based on dynamics.

6.5. On the impact of transcription errors

In addition to the previous results, it is interesting to evaluate how the transcription characteristics, mainly the error rate, influences the segmentation accuracy. Contradictory results exist in the literature: on the one hand, Christensen et al. (2005) state that transcription errors have little effect on the performance of their segmentation system. On the other hand, Huet et al. (2008) show a large gap of performance between manual and automatic transcripts on topic segmentation of radio broadcasts. This apparent contradiction can be explained by the fact that the technique used by Christensen et al. (2005) is supervised and based on discourse markers detection, while the approach of Huet et al. (2008), as ours, is unsupervised and based on lexical cohesion. This latter approach thus heavily relies on the quality of the whole transcript.

This transcript quality varies according to the ASR systems and the collections considered. In our case, the word error rates are about 30% in the good acoustic conditions of news shows, but can reach 70% for talk shows and debates where noisy environment and overlapping speeches of multiple speakers make the automatic recognition difficult. These ASR related problems are of course not specific to the collections that we use, and some authors have proposed different techniques to overcome the difficulties caused by transcription errors or oral specifics. For instance, some studies suggest to make the most of features specific to spoken documents in addition to lexical cohesion, like speaker recognition of the anchor speaker (Amaral and Trancoso, 2003), or prosody (Stolcke et al., 1999). Yet in practice, as Guinaudeau et al. (2012) noted it, these indices are seldom used for the automatic

extraction since such information is difficult to obtain and often requires document-specific knowledge.

In the two next experiments, we aim to evaluate the influence of transcription errors on our system. In particular, we measure the sensibility of the similarity functions to these errors. As a first experiment, a subset of the TV news collection was transcribed with another text-to-speech system developed by LIMSI-CNRS (Gauvain et al., 2002). This transcription system performs well on this type of document for which it has been optimized: its word error rate evaluated on this corpus is 30.4% while the aforementioned system IRENE obtains 36.1%.

As one can see in Tab. 3, this gain has a favorable effect on the segmentation results, whatever the system considered. But it is interesting to note that some systems are more dependent on the transcription quality. Indeed, the TF-IDF based system yields a large improvement both for F1 and WindowDiff criteria; the gain of Okapi is solid yet lower. In comparison, the Vectorization results are still the best, but are only slightly improved on the LIMSI transcriptions.

Table 3: Performance of topic segmentation systems on the *News* corpus according to the transcription system used

Methods	IRENE		LIMSI	
	F1	WD	F1	WD
Utiyama and Isahara (2001)	59.44	-	62.15	-
Guinaudeau et al. (2010)	61.7	-	63.7	-
Watershed + TF-IDF	49.40	0.3421	55.59	0.2977
Watershed + Okapi	60.04	0.2571	63.54	0.2288
Watershed + Vectorization	**69.07**	**0.2269**	**71.16**	**0.2226**

To push further this analysis, Tab. 4 presents the results obtained with a subset of the TV news collection that were manually transcribed. Note that since this subset is composed of only 8 of the 60 broadcasts of the collection, the results used as comparison slightly vary from the one presented in Tab. 1.

Here again, Vectorization appears as less sensitive to transcription errors: the gain (in F1-score or WindowDiff) is solid but less important than for the TF-IDF or Okapi-based similarities. This result shows the same tendency

Table 4: Performance of topic segmentation systems on a manually transcribed subset of the *News* corpus

Methods	IRENE		Manual	
	F1	WD	F1	WD
Utiyama and Isahara (2001)	65.56	-	72.96	-
Guinaudeau et al. (2010)	68.94	-	73.30	-
Watershed + TF-IDF	60.73	0.2854	71.35	0.1978
Watershed + Okapi	63.38	0.2702	70.58	0.2075
Watershed + Vectorization	**69.44**	**0.2096**	**73.66**	**0.1851**

than the previous experiments and demonstrates that the indirect similarity computation of the vectorization makes our approach more robust to noisy data from the automatic transcription process. Yet, this advantage is less important when dealing with high quality transcripts.

7. Conclusion

In this paper, we have proposed a topic segmentation algorithm used to segment transcribed TV streams. It is based on the watershed transform, a mathematical morphology tool commonly used for image segmentation; it may be seen as an approach superseding the seminal TEXT-TILING tool (Hearst, 1997). Yet, beyond this tool, we have shown that the key component is the gradient calculus, that is, the way the similarity between utterances is computed. In particular, the TF-IDF approach, still in use in many studies, shows lower results than more modern yet standard similarity computation techniques such as Okapi-BM25. Using our Vectorization principle even outperforms Okapi; indeed, this way to compute indirect similarity measures allows us to tackle the small-segment problem and the noisy data produced by ASR. The experiments reported emphasize the interest of such an approach, especially when the ASR system is error-prone.

Many developments can be foreseen for this study. As it was already identified by several authors (Merlino et al., 1997; Zhai et al., 2005; Poulisse and Moens, 2009; Guinaudeau et al., 2012; Dumont and Quénot, 2012), additional clues are available in real-life applications. Indeed, ASR systems also provide relevant pieces of information that may be used to further improve

the segmentation task, such as prosody marks or confidence measures. As it was mentioned in Sec. 2, in a multimodal setting, content-based features extracted from the video (shot detection, face recognition, overlaid texts...) could also be exploited.

From a more technical point-of-view, the image to text (or speech) analogy can be pushed further. Many improvements of the watershed and other approaches were proposed for image segmentation. We foresee their adaptation to our topic segmentation problems. In particular, hierarchical morphological segmentation schemes would be of great interest in our stream indexing framework in order to obtain a multiscale topic segmentation result.

8. Acknowledgments

This work was partially funded by OSEO, French state agency for innovation, in the framework of the Quaero project (`www.quaero.org`), and by Inria (`www.inria.fr`).

We would like to thanks our colleagues C. Guinaudeau, G. Gravier and P. Sébillot (IRISA, France), as well as L. Sitbon and P. Bellot (LIA, France) for making their datasets available for our experiments, and for the seminal discussion about TV stream segmentation.

References

Abraham, I., Bartal, Y., Neiman, O., 2006. Advances in metric embedding theory. In: Proceedings of Symposium on Theory Of Computing. Seattle, USA.

Amaral, R., Trancoso, I., 2003. Topic indexing of TV broadcast news programs. In: Proceedings of the 6th International Workshop on Computational Processing of the Portuguese Language. Faro, Portugal.

Amir, A., Argillander, J. O., Berg, M., Chang, S.-F., Franz, M., Hsu, W., Iyengar, G., Kender, J. R., Kennedy, L., Lin, C.-Y., Naphade, M., Natsev, A. P., Smith, J. R., Tesic, J., Wu, G., Yan, R., Zhang, D., 2004. IBM research TRECVID-2004 video retrieval system. In: Proc. of TRECVID 2004 Workshop.

Beeferman, D., Berger, A., Lafferty, J., 1999. Statistical models for text segmentation. Machine Learning 34 (1-3), 177–210.

Berry, M., Martin, D., 2005. Principal component analysis for information retrieval. In: Kontoghiorghes, E. (Ed.), Handbook of Parallel Computing and Statistics. Statistics: A Series of Textbooks and Monographs. pp. 399–413.

Blei, D. M., Ng, A. Y., Jordan, M. I., Lafferty, J., 2003. Latent dirichlet allocation. Journal of Machine Learning Research 2, 993–1022.

Carbonell, J. G., Yang, Y., Frederking, R. E., Brown, R. D., Geng, Y., Lee, D., 1997. Translingual Information Retrieval: A Comparative Evaluation. In: Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI). Nagoya, Japan.

Choi, F. Y. Y., 2000. Advances in domain independent linear text segmentation. In: Proceedings of the 1st metting of the North American Chapter of the Association for Computational Linguistics. Seattle, USA.

Christensen, H., Kolluru, B., Gotoh, Y., Renals, S., 2005. Maximum entropy segmentation of broadcast news. In: Proceedings of the 30th IEEE ICASSP. Philadelphia, USA.

Claveau, V., Lefèvre, S., 2011. Topic Segmentation of TV-streams by mathematical morphology and vectorization. In: Proceedings of InterSpeech Conference. Firenze, Italia, pp. 1105–1108.

Claveau, V., Tavenard, R., Amsaleg, L., 2010. Vectorisation des processus d'appariement document-requête. In: 7e conférence en recherche d'informations et applications, CORIA'10. Sousse, Tunisia, pp. 313–324.

Datar, M., Immorlica, N., Indyk, P., Mirrokni, V., 2004. Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the 20th ACM Symposium on Computational Geometry. Brooklyn, New York, USA.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R., 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science.

Derivaux, S., Forestier, G., Wemmert, C., Lefèvre, S., 2010. Supervised segmentation using machine learning and evolutionnary computation. Pattern Recognition Letters 31 (15), 2364–2374.

Dumont, E., Quénot, G., 2012. Automatic story segmentation for TV news video using multiple modalities. International Journal of Digital Multimedia Broadcasting 2012, article ID 732514, 11 pages, doi:10.1155/2012/732514.

Ferret, O., 2009. Improving text segmentation by combining endogenous and exogenous methods. In: 7th International Conference on Recent Advances in Natural Language Processing (RANLP 2009). Borovets, Bulgaria.

Gauvain, J.-L., Lamel, L., Adda, G., 2002. The LIMSI broadcast news transcription system. Speech Communication 37 (1-2), 89–108.

Gonzalez, R., Woods, R., 2008. Digital Image Processing, 3rd Edition. Prentice Hall.

Guinaudeau, C., Gravier, G., Sébillot, P., 2010. Improving ASR-based topic segmentation of TV programs with confidence measures and semantic relations. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association, Interspeech'10. Makuhari, Japan, pp. 1365–1368.

Guinaudeau, C., Gravier, G., Sébillot, P., 2012. Enhancing lexical cohesion measure with confidence measures, semantic relations and language model interpolation for multimedia spoken content topic segmentation. Computer, Speech and Language 26 (2), 90–104.

Hearst, M., 1997. Text-tiling: segmenting text into multi-paragraph subtopic passages. Computational Linguistics 23 (1), 33–64.

Hofmann, T., 1999. Probabilistic latent semantic indexing. In: Proceedings of the SIGIR conference. Berkeley, USA.

Huet, S., Gravier, G., Sébillot, P., 2008. Un modèle multi-sources pour la segmentation en sujets de journaux radiophoniques. In: Proceedings of the 15e Conférence sur le Traitement Automatique des Langues Naturelles.

Huet, S., Gravier, G., Sébillot, P., October 2010. Morpho-syntactic post-processing with n-best lists for improved French automatic speech recognition. Computer Speech and Language 24 (4), 663–684.

Kan, M.-Y., Klavans, J. L., McKeown, K. R., 1998. Linear segmentation and segment significance. In: Proceedings of the 6th International Workshop of Very Large Corpora (WVLC-6).

Lejsek, H., Asmundsson, F., Jónsson, B., Amsaleg, L., 2008. NV-tree: An efficient disk-based index for approximate search in very large high-dimensional collections. IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (5), 869–883.

Merlino, A., Morey, D., Maybury, M., 1997. Broadcast news navigation using story segmentation. In: Proceedings of the fifth ACM international conference on Multimedia. MULTIMEDIA '97. ACM, New York, NY, USA, pp. 381–391.

Misra, H., Hopfgartner, F., Goyal, A., Punitha, P., Jose, J., 2010. TV news story segmentation based on semantic coherence and content similarity. Lecture Notes in Computer Science 5916, 347–357.

Mulbregta, P. V., Carp, I., Gillick, L., Lowe, S., Yamron, J., 1999. Segmentation of automatically transcribed broadcast news text. In: Proceedings of the DARPA Broadcast News Workshop. Herndon, Virginia, USA.

Najman, L., Schmitt, M., 1994. Watershed of a continuous function. Signal Processing 38, 99–112.

Ostendorf, M., Favre, B., Grishman, R., Hakkani-Tür, D., Harper, M., Hillard, D., Hirschberg, J. B., Ji, H., Kahn, J. G., Liu, Y., Matusov, E., Ney, H., Shriberg, E., Wang, W., Wooters, C., 2008. Speech segmentation and spoken document processing. Signal Processing Magazine 25 (3), 59–69.

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

Poulisse, G., Moens, M.-F., 2009. Multimodal news story segmentation. In: Proceedings of the First International Conference on Intelligent Human Computer Interaction (IHCI 2009),. Springer, pp. 95–101.
URL `https://lirias.kuleuven.be/handle/123456789/211007`

Reynar, J. C., 2000. Topic segmentation: Algorithms and applications. Ph.D. thesis, University of Pennsylvania.

Robertson, S. E., Walker, S., Hancock-Beaulieu, M., 1998. Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive. In: Proceedings of the 7$^{\text{th}}$ Text Retrieval Conference, TREC-7. pp. 199–210.

Roerdink, J. B., Meijster, A., 2001. The watershed transform: Definitions, algorithms and parallelization strategies. Fundamenta Informaticae 41, 187–228.

Rosenberg, A., Hirschberg, J., 2006. Story segmentation of brodcast news in english, mandarin and arabic. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. NAACL-Short '06. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 125–128.

Salton, G., 1975. A Theory of Indexing. Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphy, USA.

Salzenstein, F., Collet, C., 2006. Fuzzy markov random fields versus chains for multispectral image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11), 1753–1767.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8), 888–905.

Sitbon, L., Bellot, P., April 2004. Adapting and comparing linear segmentation methods for French. In: 7th International Conference RIAO. Avignon, France, pp. 623–637.

Stein, B., 2007. Principles of hash-based text retrieval. In: Proceedings of the SIGIR conference. Amsterdam, Netherlands.

Stokes, N., Carthy, J., Smeaton, A. F., 2002. Segmenting broadcast news streams using lexical chains. In: Proceedings of STAIRS 2002 - STarting Artificial Intelligence Researchers Symposium. Lyon, France.

Stolcke, A., Shriberg, E., Hakkani-Tür, D., Tür, G., Rivlin, Z., Sönmez, K., 1999. Combining words and prosody for automatic topic segmentation. In: Proceedings of the DARPA workshop on Broadcast News. Herndon, Virginia, USA.

Utiyama, M., Isahara, H., 2001. A statistical model for domain-independent text segmentation. In: Proceedings of the 9th conference of the ACL. Toulouse, France.

Vempala, S., 2004. The Random Projection Method. Vol. 65 of Discrete Mathematics and Theoretical Computer Science. AMS.

Vincent, L., Soille, P., 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (6), 583–598.

Zhai, Y., Yilmaz, A., Shah, M., 2005. Story segmentation in news videos using visual and text cues. In: Proceedings of the 4th international conference on Image and Video Retrieval. CIVR'05. Springer-Verlag, Berlin, Heidelberg, pp. 92–102.