Enriching Textual Documents with Timecodes from Video Fragments

van der Sluis, Ielka; de Jong, Franciska

Published in:
Proceedings of RIAO'2000

Publication date:
2000

Enriching Textual Documents with Time-codes from Video Fragments

Ielka van der Sluis
University of Tilburg
Warandelaan 2
5037AB Tilburg, Netherlands
I.F.vdrSluis@kub.nl

Franciska de Jong
University of Twente/CTIT
P.O. Box 217
7500 AE Enschede, Netherlands
fdejong@cs.utwente.nl

Abstract

The OLIVE project aims the development of a multilingual indexing tool for broadcast material based on speech recognition, which automatically produces indexes from the sound track of a program (television or radio). Such a tool allows multimedia archives to be searched by keywords and corresponding fragments to be retrieved. This paper gives a report on the alignment module, which is one of the components of the retrieval environment to be developed in OLIVE. It assigns time-codes to non-time-coded textual documents that are describing the content of the video. Timecoding of these textual documents will increase the overall level of disclosure. Basis for the assignment is some similarity measure between the non-timed-coded texts and subtitle files or the transcripts from speech recognition. The core of the alignment module is a generic algorithm for generating the links that are the basis for the insertion of time-codes into non-time-coded texts. An additional step combines similarity values with locality information. The data used during testing are closed-caption files of Dutch news-broadcasts and the autocue files of these broadcasts. Adaptations to the initial algorithm for which improved perfomance figures were found involved a threshold related to the sentencelength, and the applications of a high- and low-frequency term stoplist compiled from the time-coded text under consideration.

1 Introduction

1.1 Automatic Time-code Generation and Multimedia Indexing

Several R&D projects aim at the development of indexing tools for broadcast material based on processing of the corresponding time-coded textual and speech material such as subtitles and soundtracks. Such a tool allows multimedia archives to be searched by keywords and corresponding fragments to be retrieved. In the video retrieval environment developed in the EU project OLIVE[1], not only time-coded material, but also non-time-coded accompanying textual documents relating to the content of a video are indexed and linked to video fragments. For this purpose a so-called alignment module has been developed which allows the automatic assignment of time-codes to non-time-coded text. As these accompanying texts are always produced manually, the quality will in general be better than the quality of the transcripts from speech recognition. Time-coding of these textual documents will therefore increase the number of successful matches of queries and indexes, and thereby the overall accessibility of the multimedia document collection.

A textual document can relate to a video in two ways:

1. one-to-one: the textual document is a complete transcription of the speech contained in the soundtrack (for example, a transcript produced for the purpose of making a subtitle)
2. otherwise: the textual document is not a transcription, but still describes the video as whole or its content (for example, a press release, a cut-list, manually produced shot-descriptions, etc.)

In the first case alignment may lead to a time-coded document that can replace the (imperfect) results from speech recognition. In the latter case the alignment will just lead to a higher number of correct

[1] OLIVE is a EU-funded project within the Telematics Application Programme, sector Language Engineering (LE-8364). Duration: Spring 1998- Summer 2000. For a more elaborate description of OLIVE, cf. De Jong, Gauvain, Hiemstra & Netter, this volume. Cf. also the project homepagehttp://twentyone.tpd.tno.nl/olive. For a general introduction into the subject of multimedia information retrieval, cf. Savino & Sebastiani, 1998 ; Maybury 1997 ; Hiemstra, De Jong & Netter 1998.

indexes. In case there is no link with a specific fragment, the a default time-code will be generated, which refers to the entire video rather than to a fragment.

This paper focuses on the description of a language independent algorithm that efficiently identifies relationships between texts and to insert time-codes. During the iterative development and testing it has been investigated at which level of granularity the performance of the alignment algorithm is optimal for the texts under consideration. It was decided two split the alignment process into two steps: in a first step it is attempt to align every sentence independent of the others and in a complementary step locality information is exploited. Evaluation of the test results indicates that this extremely simple approach can have added value for the video retrieval application area. On the other hand the results also make clear that major improvement of the performance is not to be expected without more additional processing steps involving more sophisticated language specific natural language processing, such as morphological and syntactic parsing and identification of semantic relationships, such as synonymy.

This introductory section gives a further description of the alignment problem, focussing on the kind of test data used as input, on the nature of the output and the need to add a postprocessing step to optimize the results. In section 2 the basic algorithm is presented and section 3 presents the complementary algorithm for the required postprocessing. Finally in section 4 will contain some concluding remarks and issues for further research.

1.2 Input data to be processed

The alignment module takes two files as input, a time-coded text and one non-time-coded text. The output is an aligned text. On the basis of the alignment the sentences of the non-time-coded free-text can be assigned the time-code of the corresponding sentences of the time-coded text. In the OLIVE project the focus is on building a demonstrator for language-based video retrieval that can serve the needs of users with a need for disclosure and retrieval of broadcast data. Because of the user requirements within the project the focus is on the disclosure of data from news programs.

During the development of the alignment module time-coded transcripts from news programs generated by speech recognition were not yet available. Therefore we used closed caption files from Dutch news-broadcasts as time-coded data. In the time-coded documents, each sentence is indexed with a time-code, which sets an interval for the sentence to be broadcast in the corresponding video-fragment. We used data from news programs broadcast at three different moments during the day.

The most extended news program, the 8 o'clock news, is used as time-coded text. From files for the 4 o'clock news and the 10 o'clock news, the time-codes were removed and these files were used as the files to be time-coded. As a result there are two related free-text documents per day (the 4 o'clock news and the 10 o'clock news), to be aligned to one time-coded document (the 8 o'clock news).

For all news-broadcasts autocue files are available which contain the text that is read out in the news-broadcasts. Ignoring the effects of last-minute changes, the news-items in autocue files have the same order as during the broadcast. In comparison to the closed-caption time-coded files, the autocue files contain longer sentences and more information that is not contained in the closed-caption files. In comparison with closed-caption data, autocue texts will have more similarity with the soundtrack. Therefore they were used to set a baseline for the evaluation. The total data set used during development consisted of 355 lines of autocue files, 123 lines of 4 o'clock news text, and 255 lines of 10 o'clock news text.

In general the sentences and subjects in the 4 o'clock news are very different from the 8 o'clock news. Most of the items in the 8 o'clock news do not appear in the 4 o'clock news. The items that do appear in the 4 o'clock news are much more briefly described. Fewer differences are found between the 8 o'clock and the 10 o'clock news. Most of the subjects in the 10 o'clock news are contained in the 8 o'clock news. Moreover in the 10 o'clock news these subjects are described at the same length as they

are described in the 8 o'clock news. The most obvious difference between the 10 o'clock and the 8 o'clock news is the news on the stock exchange, which only shows up in he former.

1.3 Output: spaghetti structure

In this section the envisaged results from the alignment of news-texts and the requirements that imposes on further processing will be investigated. Since for the 8 o'clock data the closed-caption files are supposed to cover the stream of spoken words, and since the spoken words are read from autocue files, the two input files are likely to have identical chronological orderings. However, for the 4 o'clock and 10 o'clock news files the correspondence to 8 o'clock news files is expected to have a spaghetti-like structure. Since the order and the content of the news-items in the 4 o'clock news programs and in the 10 o'clock programs differs from the 8 o'clock news, the resulting aligned closed-caption files will not contain the assigned time-codes in a chronological order. Moreover, for news files there are at least two possible levels of alignment, the news-item level and the sentence level. The disturbance of chronological order is likely to occur at both levels. In Figure 1, an example is given of how the alignments in an aligned closed-caption news-text may relate to the time-coded text to which they are aligned.

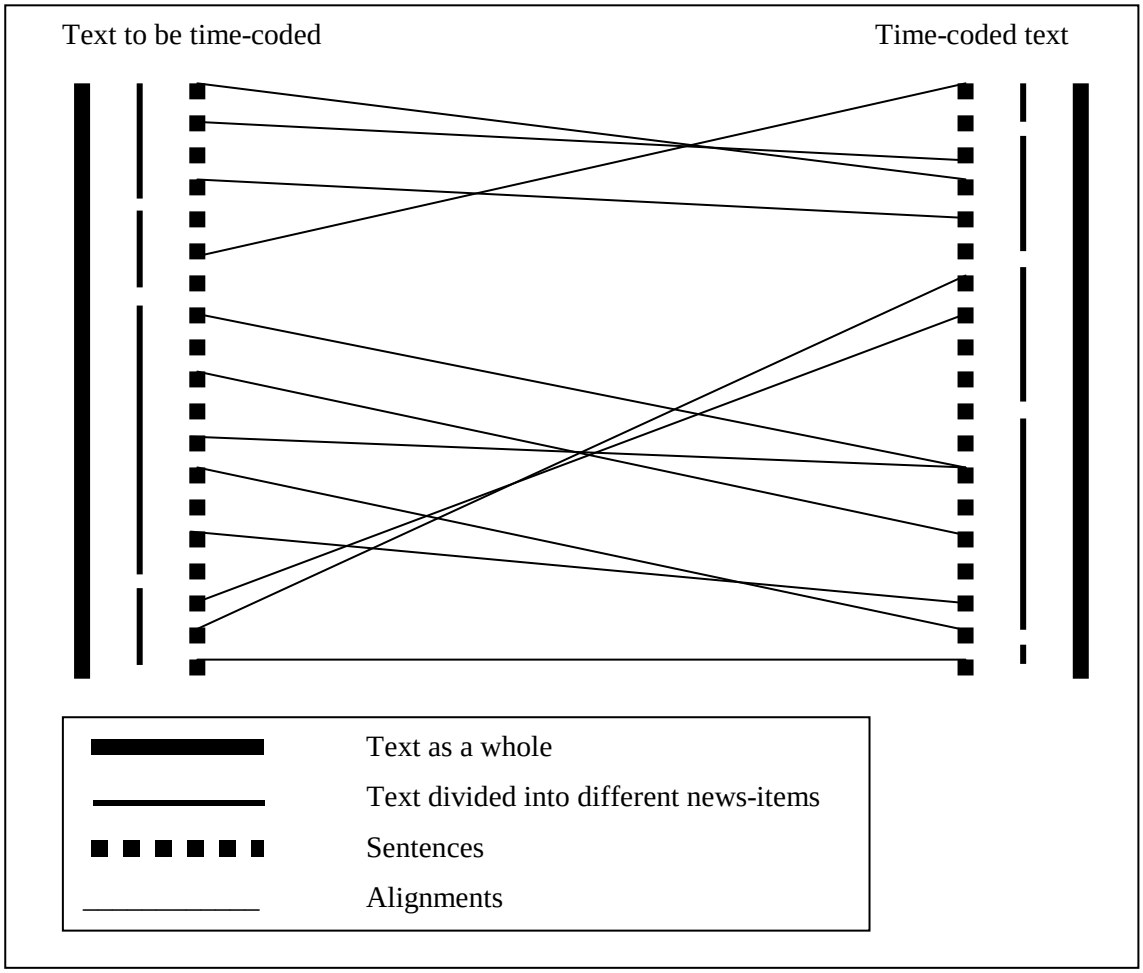

Figure 1: Example of how the sentences in a non time-coded news may relate to the time-coded text

2 The Basic Alignment Algorithm

2.1 Methodology

The starting point in the iterative development of the alignment algorithm was a very basic algorithm, using several statistical and heuristics based measures, e.g. character frequencies, word frequencies and stoplists. As these measures are either inherently non-linguistic or can be compiled automatically fom the time-coded text, no language-specific resources are required. (Since the Olive system is a multilingual indexing and retrieval environment that will support querying in German, English, French and Dutch, the alignmnet module should be able to process texts in these four languages, and therefore language independency is an issue.) Via performance tests it was determined at which level of granularity alignment is optimal and which extensions and variations are necessary to increase the performance of the algorithm. For evaluation of the results the standard information retrieval measures recall, precision and F-measure (cf. Salton & McGill 1983) were used, which for the purpose of alignment evaluation were redefined as:

$$\text{Precision} = \frac{\text{correct alignments}}{\text{correct + wrong alignments}}$$

$$\text{Recall} = \frac{\text{correct alignments}}{\text{alignments which should have be made}}$$

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision + Recall}}$$

The F-measure is taken as a criterion for the overall performance of the alignment algorithm. The fewer the number of wrong alignments and/or the number of missing alignments, the higher the F-measure. For the calculation of the recall value the test data collection has been manually processed to identify the sentences in the non time-coded files that should be aligned with sentences in the time-coded text files (65 out of 123 for the 4 o'clock news; 111 out of 255 for the 10 o'clock news; 355 out of 355 for the 8 o'clock autocuefiles.)

2.2 Procedure

The alignment algorithm aligns two texts at sentence level. This choice for this level was given in by the fact that in the input data the time-coded documents (closed caption files) have a time-code per sentence. (For a discussion of other types of alignment, cf. Church 1993 ; Gale and Church 1993 ; Owen 1998). The similarity between sentences is basically determined on the basis of the number of similar words. The algorithm makes use of an automatically compiled stoplist with high-frequency terms (e.g. function terms) to reduce their effect on the the similarity score. If there is no pair of sentences with a similirity above a certain threshold (cf. below), then the non time-coded free text is assigned the default time-code which in case of a hit at retreival time would have the consequence that the entire video rather than a specific fragment would be returned as relevant.

Below three functions are presented in which this algorithm is resumed. In 1, for each of the lines f_j in the free-text F, the alignment score (or similarity score) is defined for all the lines t_j in the time-coded text T as the sum of all the occurrences of m in t_j, for which it holds that the m^{th} word in t_j is the same as the k^{th} word in f_j. In 2, to all the lines f_j in the free-text F, i is assigned as a bestmatch, where i is the i^{th} line within the time-coded text T and where t_i has the highest alignment score.
In 3, to all the lines f_j in the free-text F the time-code of the bestmatch of f_j is assigned.

1. $$\text{alignment\_score}(t_i, f_j) = \sum_{k=1}^{N} \#\{ m \mid f_{jk} = t_{im}, 1 \le m \le \text{length}(t_l) \}$$

   where $N = \text{length}(f_j)$

2. $$\text{bestmatch}(f_j) = \underset{1 \le i \le T}{\text{argmax}} (\text{alignment\_score}(t_i, f_j))$$

3. $$\text{code}(f_j) = \text{code}(\text{bestmatch}(f_j))$$

2.3 Threshold

The threshold used by the alignment algorithm aligns the sentences dependent on their sentence length: the number of words in the sentence. After testing different values, it was found that the performance of the algorithm is best, when an alignment is only made when at least 15% of the relevant words in the free-text line is found in a time-coded-text line. When there are more sentences in the time-coded document in which this is the case, the sentence with the highest similarity score is chosen to align the free-text line to.

The threshold is defined in function 4 where each line f_j in the free-text F can be aligned only if the alignment score of t_i is more than 15 % of all the words in f_j and where i is the bestmatch of f_j. In 5, function 3 (section 2.2) is altered according to the defined threshold: to all the lines f_j in the free-text F the time-code of the bestmatch of f_j is assigned only if f_j is aligned.

4. $$\text{aligned}(f_j) = \begin{cases} \text{true}, & \text{if alignment\_score}(t_i, f_j) \ge 0.15 \cdot N \\ \text{false}, & \text{otherwise} \end{cases}$$

   where $N = \text{length}(f_j)$

   $i = \text{bestmatch}(f_j)$

5. $$\text{code}(f_j) = \text{code}(\text{bestmatch}(f_j)) \quad \text{if} \quad \text{aligned}(f_j)$$

2.4 Performance of the Basic Alignment Algorithm

In this section the performance of the best alignment method, i.e. the one using a high- and low-frequency stoplist, is presented in more detail. The method performed best on both the closed-caption and the autocue files. Braschler and Schäuble (1998) make use of medium frequency terms in the cross-language alignment of comparable texts. By not only eliminating the high frequency terms but also eliminating the low frequency terms, which are likely to give pure random associations, the list of the eliminated words increases and the number of word comparisons to be made between the time-coded and the free text decreases. Table 1 presents the recall, precision and F-measure of this algorithm. Although the precision is very high, the recall indicates that there are several missed alignments.

	Recall	**Precision**	**F-measure**
autocue files	0.51	0.98	0.67
closed-caption files	0.68	0.95	0.79

Table 1: Results for the Basic Alignment Algorithm

2.5 Concluding Remarks

After a series of iterative development phases it turned out that the of use a high- and low-frequency term stoplist yielded the best results and it was decided to take this as a starting point for the further

future development of the alignment module. The low recall indicates that in the free-text, a lot of sentences could not be aligned to the time-coded text and incorrectly are assigned a default time-code. In the next section it will be described how the performance can be improved if sentences are not just taken in isolation, but if locality information is taken into acount.

3 Exploiting locality information

3.1 Unraveling Spaghetti

The alignment results for the basic algorithm show that a lot of sentences could not be aligned. Also there is a number of incorrect alignments.

Allan (1996) describes an automatic method for gathering documents for a hypertext and linking the documents of related subject matter, based upon information retrieval techniques. An extension of this method, inspired by document visualisation and graph simplification, allows automatic classification of the links into several types from a pre-specified taxonomy of links, giving a description of the relationship between the source and the destination of the link. By graph simplification, a result of merging the links found among documents, the good and strong links between parts of the documents remain. Except for a threshold value to control this merging process, Allan also defines link edge relationships between these remaining links like, parallel, wedge, askew and crossing links.

In applying these an similar ideas to OLIVE, the following observations are crucial: in the aligned o'clock and the aligned-10 o'clock news-texts there are at least two levels of alignment: the alignment corresponding to a division of the texts into different news-items, and the alignment within the individual news-items themselves. As shown in Figure 1, the alignments have a spaghetti-like structure on both levels. The parallel, wedge, askew, and crossing alignments appear because of the alteration of the different news-items, and the alteration of the corresponding sentences within the discussion of the news-items between the input texts. This effect does not occur in the alignment of the autocue files. Here the spaghetti does only occur within news items. The ordering of the news-items themselves is identical for the caption text and autocue files. The spaghetti within news-items is due to the differences in wording and sentence length between autocue files and the 8 o'clock news captions.

Inspired by the graph simplification method, this spaghetti-problem can partly be solved by determining the strong alignments within the aligned texts. For each line in the aligned text a time-coded interval can be distinguished by taking the time-codes of the nearest strong alignment previous to and after the aligned line. A rough specification of the different news-items within the news-text can be found by assigning these time-coded intervals using a high threshold value. The time-coded intervals can be used to assign time-codes to the lines that are not yet aligned, but also incorrect alignments can be detected.

An aditional technique is needed to resolve the spaghetti pattern that occurs within news-items. For each sentence of the aligned text, the time-codes of the preceding and following line can be used to determine a time-code for an unaligned sentence, provided that the surrounding sentences fall within a specified time-interval. Below we will discuss how we implemented this second step.

3.2 Complementary Algorithm

The complementary algorithm first determines a marked interval. Then, if possible, the unaligned lines are linked to a time-coded sentence and assigned the time-codes of the surrounding lines.

3.2.1 Marked Interval

For each line in the aligned text a time-coded interval can be determined by taking the time-codes of the nearest strongly aligned lines. A rough specification of the different news-items within the news-text can be found by assigning these time-coded intervals using a high threshold value. The time-

coded intervals are used to determine if the aligned lines in the aligned-text are aligned correctly, and therefore can be used to align nearby lines which are not aligned.

The higher and lower bounds of the time-coded intervals are determined by using a threshold which relates the alignment score to the sentence length. According to the algorithm presented in section 2.2 the lines in the free-text are aligned only if at least 15% of the relevant words in the free-text line is found in a time-coded-text line. Since the threshold to be set to determine the time-coded intervals in the aligned text will be of no use if all the aligned sentences can be used as a bound of the time-coded interval, this threshold should be higher than 15 %. After testing different values, it was found that the performance of the complementary algorithm is best, when the bounds of the time-coded intervals are set using time-codes of the sentences with an alignment score higher than 20% of the sentence length. In this way free-text lines for which at least 20% of the relevant words is found in the time-coded-text line to which they are aligned, are used as the bounds of a time-coded interval.

In function 6, an interval is defined, in which for each line f_j the lower bound of the interval n is assigned the time-code of the k^{th} line in F where f_k is the nearest line previous to f_j, for which the alignment score is higher than 20% of the length of f_k . In cases where the line f_k or f_j is the first sentence of the aligned text, the lower bound of the interval is assigned the time-code f_1.

For each line f_j the higher bound of the interval m is assigned the time-code of the l^{th} line in F, where f_l is the nearest line after f_j, for which the alignment score is higher than 20% of the length of f_l. In cases where the line f_l or f_j is the last sentence of the aligned text, the higher bound of the interval is assigned the time-code of f_F.

6. $\text{interval}(f_j) = [\ \text{code}(n), \text{code}(m)\ ]$

where $n = \max\{\, k \mid (0.2 \cdot \text{length}(f_k)) \le \text{alignment\_score}(t_i, f_k) \ \wedge\ k < j \,\}$
$m = \min\{\, l \mid (0.2 \cdot \text{length}(f_l)) \le \text{alignment\_score}(t_i, f_l) \ \wedge\ l > j \,\}$
$i = \text{bestmatch}(f_j)$

3.2.2 Extra Alignments

For all unaligned lines, first the applicability of time-code of the preceding line is considered. Provided that the line preceding the line under consideration is aligned and its time-code falls within the time-coded interval, the time-code from this interval applies. The time-code of the the following line in the time-coded text is assigned to the line under consideration only if this time-code falls within the marked interval and if it is not the same as the time-code of the line after the line in the aligned text. In other cases the time-code of the aligned-line previous to the line in the aligned text applies.

For all unaligned lines for which the preceding lines are not aligned either, the time-code of the next line in the aligned text can be used under certain conditions. Provided that the line after the line under consideration is aligned and its time-code falls within the time-coded interval, the time-code from the this interval applies. The time-code of the preceding line is assigned to the line under consideration only if this time-code falls within the marked interval. If other cases the time-code of the aligned-line after the line in the aligned text applies.

7. $\text{extra\_aligned}(f_j) = \begin{cases} \text{true}, & \text{if previous}(f_j) \ \vee\ \text{next}(f_j) \\ \text{false}, & \text{otherwise} \end{cases}$

where $\text{previous}(f_j) = \begin{cases} \text{true}, & \text{if } \neg\text{aligned}(f_j) \wedge \text{aligned}(f_{j-1}) \wedge \text{code}(f_{j-1}) \in \text{interval}(f_j) \\ \text{false}, & \text{otherwise} \end{cases}$

where $\text{next}(f_j) = \begin{cases} \text{true}, & \text{if } \neg\text{aligned}(f_j) \wedge \neg\,\text{aligned}(f_{j-1}) \wedge \text{aligned}(f_{j+1}) \wedge \text{code}(f_{j+1}) \in \text{interval}(f_j) \\ \text{false}, & \text{otherwise} \end{cases}$

8. extra_code(f_j) = code(bestmatch((f_{j-1}) +1)) if previous(f_j) $\wedge$ code(bestmatch((f_{j-1}) +1)) $\in$ interval(f_j) $\wedge$ code(bestmatch((f_{j-1}) +1)) $\neq$ code(bestmatch(f_{j+1}))

9. extra_code(f_j) = code(bestmatch(f_{j-1})) if previous(f_j) $\wedge$ (code(bestmatch((f_{j-1}) +1)) $\notin$ interval(f_j) $\vee$ code(bestmatch((f_{j-1}) +1)) = code(bestmatch(f_{j+1})))

10. extra_code(f_j) = code(bestmatch((f_{j+1}) -1)) if next(f_j) $\wedge$ code(bestmatch((f_{j+1}) -1)) $\in$ interval(f_j)

11. extra_code(f_j) = code(bestmatch(f_{j+1})) if next(f_j) $\wedge$ code(bestmatch((f_{j+1}) -1)) $\notin$ interval(f_j)

3.3 Performance of the Complementary Algorithm

This section presents the performance results of the algorithm that generates the extra alignments. In the evaluation the recall and precision values and the F-measure values for the basic algorithm presented in Table 1, are compared to the values for recall and precision and the F-measure values of the complementary algorithm presented in Table 2. For both the closed-caption files and the autocue files the recall increased and the precision decreased. Although the F-measure increased, for both the closed-caption and the autocue files the complementary algorithm causes more sentences to be aligned incorrectly.

	Recall	**Precision**	**F-measure**
closed-caption files	0.92	0.76	0.83
autocue files	0.81	0.95	0.87

Table 2: Results for the Complementary Algorithm

3.4 Concluding Remarks

The results of the complementary algorithm indicate that the alignment on both the autocue and the closed-caption files improved, in comparison to the results from the basic alignment algorithm. However for both the closed-caption and the autocue files the complementary algorithm causes more sentences to be aligned incorrectly. In aligned text with a default time-code the complementary algorithm generates links for the sentences surrounding aligned sentences. This way each incorrectly aligned sentence surrounded by sentences which are not aligned, results in two more incorrect alignments. A variation to avoid this effect was implemented and tested, however the resulting F-measure did not increase. In the further development of the alignment algorithm it should be tested in what way the incorrectly aligned sentences can be detected and altered or removed.

Section 4 contains a summary of the results and an overview of issues for future research.

4 Conclusion

4.1 Summary

For the development of the alignment module, several statistics- and heuristics-based approaches were tested using, for example, character frequencies, word frequencies and stop lists. In a first evaluation, autocue files referring to the same programs as the subtitle files were aligned to the time-coded subtitle files. The autocue files in these tests serve as near perfect manual transcriptions (to replace the results of speech recognition that were not available at the time of the evaluation). The evaluation showed that an average performance of 98 % precision and 51 % recall on manually aligned test data. The use of additional heuristics that look at surrounding alignments could improve the recall up to 81 % reducing the precision only to 95 %.

In a second test the time-codes were removed from closed caption files belonging to news programs that were broadcast on the same day as the programs of the time-coded subtitle files. The resulting non-time-coded subtitle files serve as related material (text pertaining to the video files to be indexed).

They belong to programs that cover a lot of the same news events, but possibly in a different order and possibly with one or two items that should not be aligned at all. On these data the basic alignment algorithm achieved a precision of 75 % and a recall of 70 % on average. The use of additional heuristics that look at surrounding alignments improved the performance results considerably to a precision of 76 % and a recall of 92 % on average.

4.2 Future Development Issues

The previous description of the alignment module centers on the development and preliminary evaluation of the alignment algorithm. Below follows an overview of future engineering and research issues, some of which may lead to oerformance improvement.

Data collection and functionality

So far the alignment algorithm has been tested on closed-caption files of Dutch news programmes, and the autocue files of these broadcasts. The performance figures on these files indicate that by alignment more terms can be related to a video-fragment. To find out whether for the user within the OLIVE consortium the alignment module leads to a higher number of correct indexes and /or retrieved documents, the alignment algorithm also has to be tested on the files provided by ARTE, INA, TROS and NOB. These are the broadcasting organisations that are the users of the OLIVE system. Therefore eventually the alignment module should be tested on the files containing the speech recognized transcripts of a video. At the moment, none of these files are available, and the algorithm is tested on autocue files instead.

Apart from the improvement of multimedia disclosure there is a another envisaged functionality of alignment module, viz. validation and improvement of the transcripts resulting from speech recognition. In the further development of the alignment module, when the speech recognized transcripts are available, it should be evaluated if with the use of the alignments made by the algorithm, the results from the speech recognizer can be validated and improved.

Multilinguality

Up till now, the alignment algorithm is tested only on Dutch texts. Since the OLIVE System will support querying in German, English, French and Dutch, the algorithm should be able to process texts in these languages. In the further testing and development phases of the algorithm, the performance of the alignment algorithm on texts in the other languages than Dutch should be evaluated. Because the alignment algorithm uses a high- and low- frequency term stoplist, compiled from the time-coded text under consideration, no language specific resources are required to align texts in other languages. However, if more sophisticated language processing techniques will be incorporated in order to increase the overall level of performance, some of the generic measures mentioned might be replaced.

XML-Compatibility

The alignment program should be made compatible to the incoming OLIVE XML – documents. Also the program should assign the appropriate XML tags to the aligned text, distinguishing the free-text parts, which could not be aligned, and the aligned parts of the text.

User Evaluation

For the user it is invisible if a video fragment has been retrieved because of the linking in of time-codes. However the user can indicate if the retrieved information matches his request. Within the OLIVE System it may be possible to mark the documents retrieved via alignment. This information may then be used to evaluate the alignment module within OLIVE.

Acknowledgements

The research reported in this paper was partly funded by the European Commission, via the OLIVE project (http://twentyone.tno.tpd.nl/olive/) .The ideas underpining the experiments presented here are partly based on discussions with Arjan van Hessen, Djoerd Hiemstra, Godfrey Smart and Thijs Westerveld, whose implicit contribution we gratefully acknowledge.

References

J.Allan. (1996), Automatic Hypertext Link Typing. In: *Proceedings of Hypertext 1996*.

M.Braschler and P.Schäuble (1998), Multilingual Information Retrieval Based on Document Alignment Techniques. In: *Proceedings of 2nd European Conference on Digital Libraries*, pp. 183-199.

K.W. Church (1993), Char_align: A Program for aligning Parallel Texts at the Character Level. In: *Proceedings of 31st Annual Meeting of the Association for Computational Linguistics*, pp.1-8.

W.A. Gale and K.W. Church (1993), A Program for Aligning Sentences in Bilingual Corpora. In: *Computational Linguistics*, vol 19, pp.75-103.

D. Hiemstra, F.M.G. de Jong & K. Netter (eds.) (1998), Language Technology in Multimedia Information Retrieval. Proceedings of the 14th Twente Workshop on Language Technology, Enschede.

F.M.G de Jong, J.-L.Gauvain, J. den Hartog, K. Netter (1999), OLIVE. Speech-Based Video Retrieval. In: *Proceedings of CMBI'99. European Workshop on Content-Based Multimedia Indexing*. Toulouse, pp.75-80.

F.M.G de Jong, J.-L.Gauvain, D. Hiemstra and K. Netter (2000), Language-Based Multimedia Information Retrieval. In this volume.

C.B. Owen (1998), Parallell Text Alignment. In: *Proceedings of 2nd European Conference on Digital Libraries*, pp. 235-261.

M. Maybury (ed.) (1997), "Intelligent Multimedia Information Retrieval", MIT Press, Cambridge.

G.Salton and M.J.McGill (1983), Introduction to Modern Information Retrieval, McGraw Hill, New York.

P. Savino and F. Sebastiani (1998), Essential bibliography on Multimedia Information Retrieval, Categorisation and Filtering.

I.F.van der Sluis (1999), Assigning Time-codes to Video-related Textual Documents, master thesis, University of Groningen.