HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Universität Potsdam

E-Lecture Material Enhancement Based on Automatic Multimedia Analysis

Dissertation

zur Erlangung des akademischen Grades
des Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachgebiet Internet-Technologien und -Systeme
des Hasso-Plattner-Instituts

eingereicht an der
Hasso-Plattner-Institut, Digital Engineering Fakultät
der Universität Potsdam

vorgelegt von M.Sc.

Xiaoyin Che

Potsdam, Juni 2017

odi et amo, sic vita est.

Dissertation Reviewers:
Prof. Dr. Christoph Meinel, Hasso-Plattner-Institut
Prof. Dr. Baocai Yin, Dalian University of Technology
Prof. Dr. Wolfgang Effelsberg, Universität Mannheim

Examination Committe:
Prof. Dr. Felix Naumann, (Chairman)
Prof. Dr. Patrick Baudisch,
Prof. Dr. Andreas Polze,
...and the reviewers

Disputation: 01.02.2018

Note: *magna cum laude*

Published online at the
Institutional Repository of the University of Potsdam:
URN urn:nbn:de:kobv:517-opus4-408224
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-408224

Declaration

I herewith declare that I have produced this thesis without the prohibited assistance of third parties and without making use of aids other than those specified. Notions taken over directly or indirectly from other sources have been identified as such. All data and findings in the work have not been falsified or embellished. This thesis has not previously been presented in identical or similar form to any other German or foreign examination board.

The thesis work was conducted from *September. 2012* to *June. 2017* under the supervision of *Prof. Dr. Christoph Meinel*.

Xiaoyin Che

Potsdam, Germany

Abstract

In this era of high-speed informatization and globalization, online education is no longer an exquisite concept in the ivory tower, but a rapidly developing industry closely relevant to people's daily lives. Numerous lectures are recorded in form of multimedia data, uploaded to the Internet and made publicly accessible from anywhere in this world. These lectures are generally addressed as e-lectures. In recent year, a new popular form of e-lectures, the _**M**assive **O**pen **O**nline **C**ourses_ (MOOCs), boosts the growth of online education industry and somehow turns "learning online" into a fashion.

As an e-learning provider, besides to keep improving the quality of e-lecture content, to provide better learning environment for online learners is also a highly important task. This task can be preceded in various ways, and one of them is to enhance and upgrade the learning materials provided – e-lectures could be more than videos. Moreover, this process of enhancement or upgrading should be done automatically, without giving extra burdens to the lecturers or teaching teams, and this is the aim of this thesis.

The first part of this thesis is an integrated framework of multi-lingual subtitles production, which can help online learners penetrate the language barrier. The framework consists of _**A**utomatic **S**peech **R**ecognition_ (ASR), _**S**entence **B**oundary **D**etection_ (SBD) and _**M**achine **T**ranslation_ (MT), among which the proposed SBD solution is major technical contribution, building on _**D**eep **N**eural **N**etwork_ (DNN) and _**W**ord **V**ectors_ (WV) and achieving state-of-the-art performance. Besides, a quantitative evaluation with dozens of volunteers is also introduced to measure how these auto-generated subtitles could actually help in context of e-lectures.

Secondly, a technical solution "TOG" (_**T**ree-Structure **O**utline **G**eneration_) is proposed to extract textual content from the displaying slides recorded in video and re-organize them into a hierarchical lecture outline, which may serve in multiple functions, such like preview, navigation and retrieval. TOG runs adaptively and can be roughly divided into intra-slide and inter-slides phases. Table detection and lecture video segmentation can be implemented as sub- or post-application in these two phases respectively. Evaluation on diverse e-lectures shows that all the outlines, tables and segments achieved are trustworthily accurate.

Based on the subtitles and outlines previously created, lecture videos can be further split into sentence units and slide-based segment units. A lecture highlighting process is further applied on these units, in order to capture and mark the most important parts within the corresponding lecture, just as what people do with a pen when reading paper books. Sentence-level highlighting depends on the acoustic analysis on the audio track, while segment-level highlighting focuses on exploring clues from the statistical information of related transcripts and slide content. Both objective and subjective evaluations prove that the proposed lecture highlighting solution is with decent precision and welcomed by users.

All above enhanced e-lecture materials have been already implemented in actual use or made available for implementation by convenient interfaces.

Zusammenfassung

In der Ära der mit Hochgeschwindigkeit digitalisierten und globalisierten Welt ist die Online-Bildung nicht mehr ein kunstvoller Begriff im Elfenbeinturm, sondern eine sich schnell entwickelnde Industrie, die für den Alltag der Menschen eine wichtige Rolle spielt. Zahlreiche Vorlesungen werden digital aufgezeichnet und im Internet Online zur Verfügung gestellt, so dass sie vom überall auf der Welt erreichbar und zugänglich sind. Sie werden als e-Vorlesungen bezeichnet. Eine neue Form der Online-Bildung namens „*__M__assive __O__pen __O__nline __C__ourses*“ (MOOCs), welche zum Trend seit dem letzten Jahr geworden ist, verstärket und beschleunigt die Entwicklung des Online-Lernens.

Ein Online-Lernen Anbieter hat nicht nur die Qualität des Lerninhaltes sondern auch die Lernumgebung und die Lerntools ständig zu verbessern. Eine diese Verbesserungen ist die Form, in der das Lernmaterial aktualisiert und angeboten wird. Das Ziel dieser Dissertation ist die Untersuchung und die Entwicklung von Tools, die der Prozess der Verbesserung und Aktualisierung des Lernmaterials automatisch durchführen. Die entwickelten Tools sollen das Lehrerteam entlasten und seine Arbeit beschleunigen.

Der erste Teil der Dissertation besteht aus einem integrierten Framework für die Generierung von mehrsprachigen Untertiteln. Dies kann den Online-Lernern ermöglichen, die Sprachbarriere beim Lernen zu überwinden. Das Framework besteht aus „*__A__utomatic __S__peech __R__ecognition*“ (ASR), „*__S__entence __B__oundary __D__etection*“ (SBD), und „*__M__achine __T__ranslation*“ (MT). SBD ist realisiert durch die Anwendung von „*__D__eep __N__eural __N__etwork*“ (DNN) und „ *__W__ord __V__ectors*“ (WV), wodurch die Genauigkeit der Stand der Technik erreicht ist. Außerdem quantitative Bewertung durch Dutzende von Freiwilligen ist also eingesetzt, um zu evaluieren, wie diese automaisch generierten Untertiteln im Kontext vom Online-Lernen helfen können.

Im zweiten Teil ist eine technische Lösung namens „ *__T__ree-Structure __O__utline __G__eneration*“ (TOG) für die Extraktion des textuellen Inhalts aus den Folien präsentiert. Der extrahierten Informationen werden dann in strukturierter Form dargestellt, welche die Outline der Vorlesung wiederspiegelt. Diese Darstellung kann verschiedenen Funktionen dienen, wie dem Vorschau, der Navigation, und dem Abfragen des Inhaltes. TOG ist adaptiv und kann grob in Intra-Folie und Inter-Folien Phasen unterteilt werden. Für diese Phasen, Tabellenerkennung und die Segmentierung von Vorlesungsvideo können als Sub- oder Post-Applikation jeweils implementiert werden. Die höhere Genauigkeit der extrahierten Outline, der Tabellen, und der Segmenten wird experimentell durch die Anwendung auf verschieden e-Vorlesungen gezeigt.

Basierend auf den Untertiteln und dem Outline, die in vorher generiert wurden, Vorlesungsvideos können weiter in Satzeinheiten und Folien-basierten Segmenteinheiten gesplittet werden. Ein Hervorhebungsprozess wird weiter auf diese Einheiten angewendet, um die wichtigsten Teile innerhalb der entsprechenden Vorlesung zu erfassen und zu markieren. Dies entspricht genau, was die Lerner mit einem Stift beim Lesen von Büchern machen. Die Satz-Level-Hervorhebung hängt von der akustischen Analyse auf der Audiospur ab, während die Segment-Level-Hervorhebung auf die Erforschung von Hinweisen aus den statistischen Informationen der verwandten Transkripte und des Folieninhalts fokussiert. Die objektiven und subjektiven Auswertungen zeigen, dass die vorgeschlagene Vorlesungsvorhebungslösung mit anständiger Präzision und von den Benutzern akzeptiert wird.

All diese Methoden für die Verbesserung der Online-Materialien wurden bereits für den Einsatz implementiert und durch komfortable Schnittstellen zur Verfügung gestellt.

Contents

List of Figures

List of Tables

Chapter 1

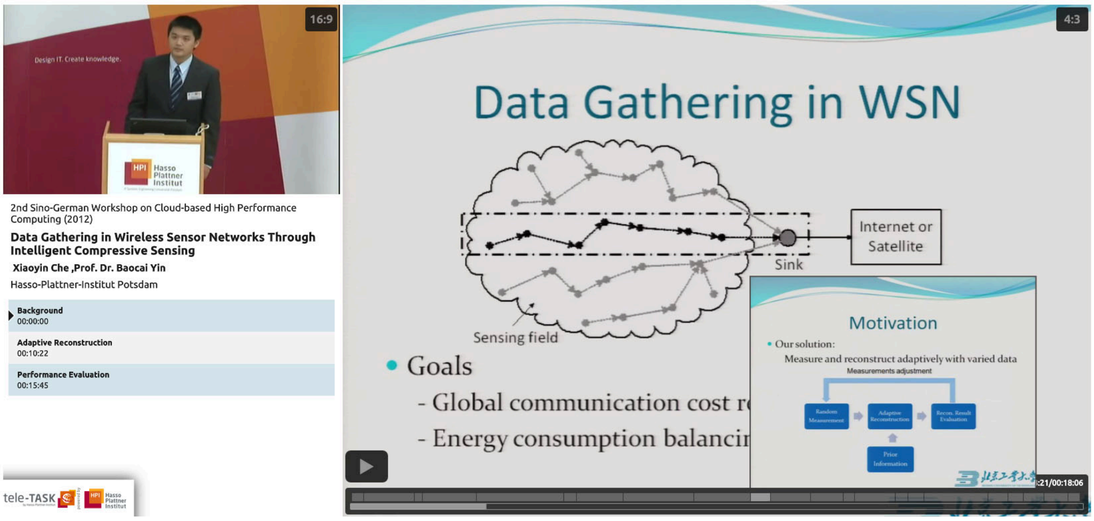
Introduction

1.1 Research Background

It's never too late to learn. Education therefore becomes a never-out-of-date research topic, driven by the desire to better spread knowledge, in which distance learning or e-learning technology is considered as a highly important aspect [1, 2]. With the rapid development of computer science and communication technology in recent decades, we have witnessed the great progress of e-learning technologies and the explosive growth of high quality e-learning resources. In this era, geometrical limitation is no longer a major barrier in knowledge spreading. Learning whenever and wherever is nowadays a dream come true.

In ***H****asso* ***P****lattner* ***I****nstitute* (HPI), lectures are recorded by tele-TASK recording system, the ***tele-T****eaching* ***A****nywhere* ***S****olution* ***K****it*, and uploaded to tele-TASK.de[1] since 2002 [3, 4]. Tele-TASK recording system applies a double-stream recording strategy, with a lecturer stream focusing on the speaker and capturing the audio signals, while when applicable, a silent desktop stream is used to record what is displaying on the speaker's laptop. The two streams are automatically synchronized, as shown in Figure 1.1. On the web portal of tele-TASK, there are in total around 5900 lectures online and in 2016, there were more than 68000 visitors. Generally, the website of tele-TASK serves as a video-based lecture archive in purpose of assisting classroom learning.

[1] https://www.tele-TASK.de



2nd Sino-German Workshop on Cloud-based High Performance Computing (2012)

Data Gathering in Wireless Sensor Networks Through Intelligent Compressive Sensing

Xiaoyin Che ,Prof. Dr. Baocai Yin

Hasso-Plattner-Institut Potsdam

Background 00:00:00

Adaptive Reconstruction 00:10:22

Performance Evaluation 00:15:45

Data Gathering in WSN

- Goals
  - Global communication cost r
  - Energy consumption balanci

Motivation

- Our solution:
  Measure and reconstruct adaptively with varied data

Figure 1.1: A screenshot of a lecture recorded in double-stream format and displayed by tele-TASK system.

In recent years, a new form of distance learning, ***M**assive* ***O**pen* ***O**nline* ***C**ourses* (MOOCs), turns "learning online" into a fashion. The term "MOOC" is believed to be first mentioned in 2008 [5] and this concept has expanded at an incredible speed, particularly in 2012, "the year of the MOOC" [6]. By now, many top universities and educational organizations are involved in the wave of MOOCs, which attract millions of learner into this community, regardless of their ages, genders, nationalities and educational backgrounds.

HPI also has its own MOOC platform, openHPI.de[1]. It launched in September 2012 and concentrates on information technology [7, 8]. OpenHPI offers typical xMOOC, according to the categories of cMOOC and xMOOC. Until the end of 2016, there were 33 courses online with the total enrolments over 300000. The courses are generally instructed in English or German, but since February 2014, 6 courses have been offered in simplified Chinese on a sub-platform openHPI.cn[2], which is hosted in Shanghai. In general, openHPI courses are recorded by tele-TASK recording system.

No matter in traditional tele-teaching or in recently popular MOOCs, the core material is undoubtedly the lecture video, in addition with slides, notes, tests,

[1] https://open.hpi.de/

[2] https://openhpi.cn/

etc., as supplementaries. From the viewpoint of an e-learning provider, these are basic course materials. Indeed, basic course materials could be sufficient to support the smooth progress of an online course, but e-learning providers should not be satisfied with it. There are still a lot of imperfections that can be and need to be improved.

For instance, it is impossible for a learner to attend a course which is instructed in a language he/she does not understand. Such a course might contain excellent content but unfortunately become transparent to this learner because of the language barrier. Even for those who have learned the course language as a foreign language in school for several years, their learning achievements might be downgraded in comparison with native or fluent speaking learners. In this case, multilingual extension of the course is highly desirable.

Since numerous lectures are recorded in videos and uploaded online for free access, it is also very challenging for a learner to find the most appropriate course, or more specifically, the certain lecture video. With just a glance at the lecture title, the learner can hardly have a clear idea about whether this lecture is exactly what he/she wants. In this case, accurate lecture description with more details should be provided.

Besides, a common concern about online lectures is the difficulty to keep online learners as engaged as those in classroom [9, 10]. Research shows that the median engagement time for MOOC learners when watching a lecture video is at most 6 minutes [11]. Since many lecture videos are way longer than that, how to help learners concentrate and further improve their learning efficiency is also open for discussion.

Apparently, basic course materials cannot meet these demands and need to be enhanced. In order to offer multilingual extension, provide detailed lecture description or keep learners more concentrating, additional course materials are needed. Let's address them as advanced course materials. However, it will be a huge extra burden for the teachers or the teaching teams if they are required to prepare these advanced course materials manually. Automation is obviously a better choice.

In this thesis, several solutions are proposed in order to prepare multiple types of advanced materials for online courses, mainly by automatically analyzing and processing multimedia data, such as lecturer stream video, desktop stream video and slides in PDF or PPTX format, which are also the basic course materials and provided in both tele-TASK and openHPI systems. For an upcoming course, the complete process can be done before it starts. Meanwhile, it is also applicable for archived lectures.

1.2 Contribution and Publications during Ph.D. study

The purpose of the techniques developed in this thesis is about how to provide better e-learning service by enhancing e-lecture materials in order to further facilitate online learners. Therefore, to understand what online learners actually want and need is the crucial prerequisite to design, develop and offer meaningful and successful solutions. The author was involved in the teaching teams of no less than 5 MOOCs on openHPI.de and openHPI.cn, which have different target user groups. These experiences gave the author good opportunities to directly interact with online learners through forum discussion or questionnaire, and to observe the similarities and differences of their behaviors according to distinguished user groups. Some of the observations and analyses are published in the following paper:

- **Che, X.**, Luo, S., Wang, C., Meinel, C.: An attempt at mooc localization for chinese-speaking users. In: International Journal of Information and Education Technology, 6(2), 90. (2016)

Throughout these interactions and observations, along with the massive literature reading and past experiences, the author aims to provide an advanced e-lecture materials combo, which includes but not limits to multi-lingual lecture subtitles, hierarchical lecture outlines, logical lecture segments and sentence-level or segment-level lecture highlights. These additional materials could enhance online learners' learning experiences with video-based e-lectures, while all of them would be generated automatically without giving extra burdens to the teachers.

In order to automatically achieve these advanced materials, three technically somehow independent but logically highly relevant solutions are proposed. As illustrated in Figure 1.2, *Lecture Subtitle Production* works with the lecturer stream

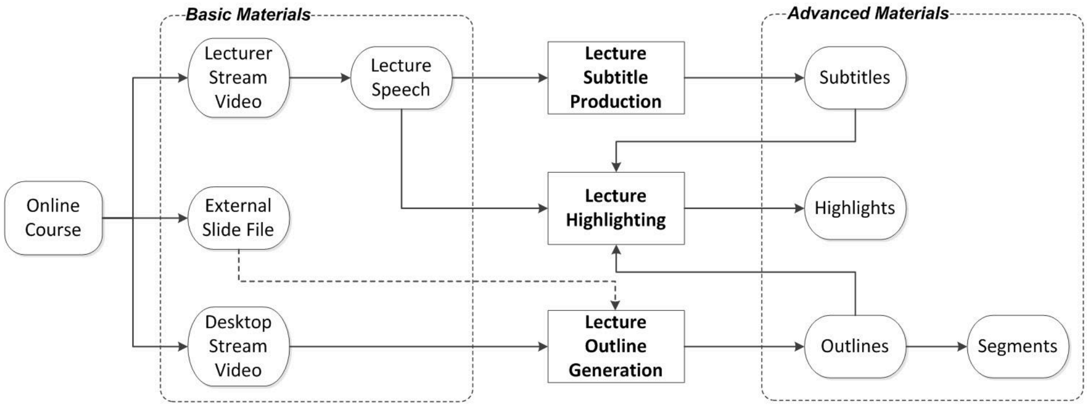


Figure 1.2: The structure of the technical solutions involved in this thesis.

video to create subtitles, while *Lecture Outline Generation* processes on the desktop stream video to extract outlines, then *Lecture Highlighting* is applied based on the outputs from previous two solutions and connects all efforts together.

In *Lecture Subtitle Production*, an integrated framework consisting of ***A**utomatic **S**peech **R**ecognition* (ASR), ***S**entence **B**oundary **D**etection* (SBD) and ***M**achine **T**ranslation* (MT) is built, in which SBD is the major technical contribution. The proposed SBD approach consists of a novel lexical model which takes ***W**ord **V**ectors* (WV) as the only feature and is constructed by a ***D**eep **N**eural **N**etwork* (DNN), a simplest but effective pause-only acoustic model, and a 2-stage joint decision scheme. In context of English, it reaches the state-of-the-art accuracy and becomes popular comparing basis within the research community. Related publications include:

- **Che, X.**, Wang, C., Yang, H., Meinel, C.: Punctuation prediction for unsegmented transcript based on word vector. In: The 10th International Conference on Language Resources and Evaluation (LREC). (2016)
- **Che, X.**, Luo, S., Yang, H., Meinel, C.: Sentence Boundary Detection Based on Parallel Lexical and Acoustic Models. In: Proceeding of Interspeech 2016, 2528-2532. (2016)

The functionality of SBD is also extended from English to German. Unlike in English SBD which applies public word vector sets with high reputation, German word vectors are trained by the author and his colleagues, and evaluated by a novel traversal-free evaluation metric, along with traditional ones. Practically, German SBD using self-trained word vectors achieved similar level of accuracy with the highly successful English SBD, which indirectly demonstrates the high quality of the German word vectors trained. Word vector related contribution is concluded in following submission:

- **Che, X.**, Raschkowski, W., Ring, N., Yang, H., Meinel, C.: Traversal-Free Word Vector Evaluation in Analogy Space. Submitted to: RepEval 2017, co-located with EMNLP 2017.

By slightly adjusting the SBD approach and implementing ASR and MT services, multi-lingual subtitles can be automatically produced for e-lectures. In order to evaluate the quality of such auto-generated subtitles, a comprehensive test with dozens of volunteers with different backgrounds is conducted. By this effort, the accuracies of both source and target language are analyzed and moreover, how these auto-generated subtitles could help the human editors in post-processing is also quantitatively measured, which has significant practical value in the field of subtitle production. These contributions are published in:

- **Che, X.**, Luo, S., Yang, H., Meinel, C.: Automatic Lecture Subtitle Generation and How It Helps. In: Advanced Learning Technologies (ICALT), 2017 IEEE 17th International Conference on. IEEE. (2017)

In *Lecture Outline Generation*, an up-to-6-level lecture outline is obtained and presented in tree-structure. It is extracted from the lecture slides which are originally recorded in video form. After executing ***S****lide* ***T****ransition* ***D****etection* (STD) and ***O****ptical* ***C****haracter* ***R****ecognition* (OCR) on the input desktop stream video, the textual content can be reconstructed per slide by adaptively analyzing the page layout and the logical relations between adjacent slides will also be explored. Finally, the hierarchical lecture outline with trustworthy precision can be achieved and it would be capable to cope with functions like preview, navigation and retrieval. Related publications include:

- **Che, X.**, Yang, H., Meinel, C.: Tree-structure outline generation for lecture videos with synchronized slides. In: e-Learning and e-Technologies in Education (ICEEE), 2013 Second International Conference on (pp. 87-92). IEEE. (2013)
- **Che, X.**, Yang, H., Meinel, C.: The Automated Generation and Further Application of Tree-Structure Outline for Lecture Videos with Synchronized Slides. In: International Journal of Technology and Educational Marketing (IJTEM), 4(1), 34-50. (2014)
- **Che, X.**, Yang, H., Meinel, C.: Adaptive e-lecture video outline extraction based on slides analysis. In: International Conference on Web-Based Learning (pp. 59-68). Springer International Publishing. (2015)

Under the framework of outline generation, there are two sub-applications: lecture video segmentation and table detection. To segment lecture video is a tricky problem with traditional segmentation methods, since there are much less scene-changes in lecture videos than in natural videos. However, during the process of outline generation, subtopics have already been explored. By simply parsing the clues from the outline, corresponding video segments with highly logical basis can be confirmed. On the other hand, table detection is a step when analyzing the page layout in intra-slide phase of outline generation process. Although it is a widely researched topic in document analysis community, no previous effort has been made dedicating to the slide images with diverse layouts. The proposed approach outperforms ABBYY FineReader, a commercial software which is considered as the champion in detecting tables, when applied on slide images. Related publications are:

- **Che, X.**, Yang, H., Meinel, C.: Lecture video segmentation by automatically analyzing the synchronized slides. In: Proceedings of the 21st ACM international conference on Multimedia (pp. 345-348). ACM. (2013)
- **Che, X.**, Yang, H., Meinel, C.: Table Detection from Slide Images. In:Pacific-Rim Symposium on Image and Video Technology (pp. 762-774). Springer International Publishing. (2015)

In *Lecture Highlighting*, the motive is to simulate a universally existing phenomenon into e-lecture context: many people are used to hold a marker to highlight something when reading a book, especially a text book. In order to highlight the video-based lecture, the video needs to be split into a sequence of units and the importance of these units needs to be quantitatively evaluated. Naturally, the sentence units can be parsed from the lecture subtitle, while slide units are reserved in lecture outline, both of which are available from previous solutions.

Sentence-level highlighting is based on acoustic analysis. The audio signal will first be deconstructed into ***V****oiced Sound*, ***U****nvoiced Sound* and ***S****ilence* (V/U/S), and then the importance of a sentence-unit could be achieved by comprehensively considering the factors such as loudness, pitch and duration. By comparing with the expert-created ground-truth, the general precision is fairly good. Some example demonstrations and user feedbacks also support this idea.

Segment-level highlighting focuses on statistical analysis. It aims to explore the characteristics within the lecture transcripts (*subtitles*), lecture slides (*outlines*) and their correlations. The segments are defined by ***S****lide* ***U****nits* (SUs) and further categorized by the type of the related slide. Different measurements are prepared for each category. Accuracy is evaluated by comparing with user-generated ground-truth and the result is quite promising.

Finally, whether there is a connection between sentence-level and segment-level highlights is also discussed. Publications about *Lecture Highlighting* are listed here:

- **Che, X.**, Staubitz, T., Yang, H., Meinel, C.: Pre-Course Key Segment Analysis of Online Lecture Videos. In: Advanced Learning Technologies (ICALT), 2016 IEEE 16th International Conference on (pp. 416-420). IEEE. (2016)

- **Che, X.**, Luo, S., Yang, H., Meinel, C.: Sentence-Level Automatic Lecture Highlighting Based on Acoustic Analysis. In: Computer and Information Technology (CIT), 2016 IEEE International Conference on. IEEE. (2016)

- **Che, X.**, Yang, H., Meinel, C.: Automatic Online Lecture Highlighting Based on Multimedia Analysis. In: Special Issue "Innovation in Technologies for Educational Computing", IEEE Transactions on Emerging Topics in Computing and IEEE Transactions on Learning Technologies. (2017, accepted, to be appeared in October)

From a practical point of view, the automatic subtitle production tool has already been applied in preparation of several MOOCs on different platforms and is highly appreciated by the corresponding teaching staffs. The outline generation tool is partially implemented with openHPI with some successful outcomes. The lecture highlights have been presented to the MOOC users in an experimental approach and received some very encouraging feedbacks.

There are some other publications during the period of Ph.D. study as co-author, which include:

- Wang, C., Yang, H., **Che, X.**, Meinel, C.: Concept-based multimodal learning for topic generation. In: International Conference on Multimedia Modeling (pp. 385-395). Springer International Publishing. (2015)
- Yang, H., Wang, C., **Che, X.**, Luo, S., Meinel, C.: An Improved System For Real-Time Scene Text Recognition. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (pp. 657-660). ACM. (2015)
- Luo, S., Yang, H., Wang, C., **Che, X.**, Meinel, C.: Action Recognition in Surveillance Video Using ConvNets and Motion History Image. In: International Conference on Artificial Neural Networks (pp. 187-195). Springer International Publishing. (2016)
- Luo, S., Yang, H., Wang, C., **Che, X.**, Meinel, C.: Real-Time Action Recognition in Surveillance Videos Using ConvNets. In: International Conference on Neural Information Processing (pp. 529-537). Springer International Publishing. (2016)

1.3 Structure of this thesis

As illustrated in Figure 1.2, there are three major technical solutions: lecture subtitle production, lecture outline generation and lecture highlighting. Each of them will be introduced in detail by a whole chapter. Since these three solutions have different goals and comparatively independent technical backgrounds, related works will also be separately presented in corresponding chapters. Therefore, the structure of this thesis is arranged as follow:

Chapter 1 gives a general introduction. Except for the elaborations of the overall research background and the thesis structure, Chapter 1 also contains a list of the author's publications during the period of Ph.D. study.

Chapter 2 provides a panorama about the topic of lecture subtitle production. By taking lecture speech as input, a technical framework, which can automatically generate bilingual subtitles, is proposed. The framework consists of ASR, SBD and MT, with SBD as the major technical contribution. Then a quantitative study on how the auto-generated subtitles could help the human editors is also implemented.

Chapter 3 focuses on lecture outline generation. It works with the desktop stream video and optionally with external slide files. Actually, the outline is extracted from lecture slides and reconstructed in tree-structure. The hierarchical structure of such outline derives from both intra-slide and inter-slides phases. Chapter 3 also includes the introduction of an extended sub-application "table detection" and a follow-up application "lecture video segmentation".

Chapter 4 depicts the motive and method of lecture highlighting. It takes the outputs from previous two solutions as input, by which all three proposed solutions are closely linked. There are in fact two approaches: sentence-level highlighting based on acoustic analysis and segment-level highlighting based on statistical analysis. The potential correlation between them will also be discussed.

At the end of this thesis, conclusion & future works, appendices, references, acronyms and acknowledgement can be found.

Chapter 2

Lecture Subtitle Production

2.1 Motivation

As already mentioned in Chapter 1, geometrical boundary of classroom or campus is no long the border of high quality education. Numerous of video-based e-lectures, no matter in form of traditional tele-teaching or recently popular MOOC, are capable to spread knowledge to every corner of this planet. However, language barrier becomes a huge rock in this wave of knowledge globalization [12]. Multilingual extension is thus highly desirable and once implemented, it could actually be very effective [13].

Although there are some other options, such as preparing parallel lecture videos in different languages or dubbing a second audio track [14], subtitles are considered as the best breaker of language barrier [15, 16]. Subtitling is also addressed as "***C**losed **C**aptioning* (CC)" in some contexts and has already been widely used in various scenarios, not only by Internet video providers like YouTube or Vimeo, but also with traditional TV service, such as ARD or ZDF in Germany [17]. Unsurprisingly, many MOOC platforms have already implemented the function to enable subtitles [18], so does openHPI.

In a survey offered with MOOC "Internetworking" on openHPI.cn[1], respondents want not only the subtitle in target language, but also in source language, with bilingual subtitle as the most welcomed option (*64 of 99*). Some other

[1] http://www.wenjuan.com/r/a2Ufqa?pid=53d0b344f7405b5c9ba5ee58&vcode=52a90f532fae475c939350beec981117, (in Chinese)

learners claim that bilingual subtitle may facilitate them to learn both the lecture content and the foreign language in a same time [19] – killing two birds with one stone. On the other hand, subtitle in source language may also help the learners with hearing impairments [20]. Therefore when provided, the subtitle quality in both target and source language are equally important.

Generally, subtitles for online lectures are created manually by volunteer groups, course teaching teams or hired staffs. It is well known that manual subtitle production has very high cost in time or/and money. So it is natural for people to look for the possibility of automation. Due to current technical conditions, auto-generated subtitles inevitably contain errors and it is controversial that whether such imperfect subtitles could be offered directly to learners. However, if the auto-generated subtitle has decent accuracy and can be taken as draft, it will definitely help the human subtitle producer. And obviously, the complexity of manual post-editing is inversely proportional to the accuracy of the auto-generated draft.

In this chapter, an integrated framework to automatically produce bilingual or multilingual subtitle will be constructed. Addressing this problem, to feed the output of ***A**utomatic **S**peech **R**ecognition* (ASR) into ***M**achine **T**ranslation* (MT) seems to be the only possible technical path. However, how to format the ASR transcript, which is also going to be the MT input, is crucial for the quality of translation: various research works report that MT could benefit from better segmentation of transcripts [21, 22, 23]. Meanwhile, the importance of proper segmentation when subtitling in source language is also widely acknowledged [24, 25]. These facts intrigue another research topic: ***S**entence **B**oundary **D**etection* (SBD).

As the name implies, SBD aims to find grammatical boundaries of sentences or clauses in the unpunctuated continuous sequence of words. Sometimes it is also addressed as punctuation prediction, since proper punctuation marks could be restored when the boundary positions are confirmed. In this chapter a state-of-the-art SBD approach will be proposed, which adopts a structure of parallel lexical and acoustic models with posterior probabilities fusion. The proposed SBD approach is the major technical contribution in this chapter and serves as the middleware between ASR and MT to form a complete subtitle producing solution.

The rest of this chapter is arranged as follow: Chapter 2.2 discusses related works. SBD is first illustrated in detail in Chapter 2.3. Then the implementation of the integrated subtitle producing framework is presented in Chapter 2.4. Chapter 2.5 elaborates the bilingual subtitle quality evaluation and how it may help. Finally a short summary will conclude this chapter.

2.2 Related Works

Undoubtedly, ASR and MT technologies are the foundation of automatic subtitle production. But in this chapter, the development of these two technologies will not be discussed, since they are not the major research topics in this thesis.

Some initial solutions in automatic subtitling or captioning simply implement the ASR and take the silence in the audio track as sentence boundaries [26, 27]. In order to achieve better subtitle quality, how to precisely segment the ASR output is gradually getting more focused [28, 29]. Quantitative analysis proves that better segmentation result could save time for manual post-editing [30]. Furthermore, the absence of punctuation marks in ASR output is also considered as a shortcoming in auto-generated subtitles [31] and efforts have been made for that [32].

By taking SBD as an independent academic topic, research works differ mainly in three aspects: features, models and structures. Features can be divided into two groups: lexical features based on textual data and acoustic features deriving from audio signals. Most of the lexical approaches takes LM scores (***L***_anguage_ ***M***_odel_), tokens, POS tags (***P***_art_-***of***-***S***_peech_), *etc.*, of several continuous words as features to train lexical models [33, 34, 35, 36]. And frequently used acoustic features include pause, pitch, energy, speaker switch and so on [37, 38, 39]. However, multi-modal approaches using both lexical and acoustic feature are more popular.

Except for several early acoustic-only SBD solutions which use heuristic thresholds to make decision, the majority of SBD models are based on machine learning. Depending on respective circumstances, ***D***_ecision_ ***T***_ree_ (DT) [37, 40], ***H***_idden_

***M**arkov **M**odel* (HMM) [39, 41], ***C**onditional **R**andom **F**ields* (CRF) [35, 42], ***S**upport **V**ector **M**achine* (SVM) [43, 44], ***D**eep **N**eural **N**etwork* (DNN) [45, 46], *etc.*, have all been involved in SBD model building, and it is hard to evaluate which is the best. When addressing structure, lexical-only and acoustic-only approaches have already been discussed previously, while the analysis of the structures of multi-modal approaches will be elaborated later in Chapter 2.3.1.

Another focus point in subtitling is the way how to implement the subtitles to the video. There are some suggested standards for "good" subtitle [47, 48] and some applications catering to these standards [49, 50]. In order to fit the item length requirement, sentence simplification or compression is also introduced [51, 52]. However, auto-generated subtitle can still not avoid errors in spite of all kinds of efforts, therefore how to facilitate the human producers in post-editing is also widely discussed [53, 54].

The last issue is whether and how the auto-generated subtitles can help. Valor Miró *et al.* reported a test from 20 lecturers who were required to transcribe Spanish lecture videos, by which 54% of working time could be saved if draft was offered [55]. De Sousa *et al.* employed an experiment with English-Portuguese translation on subtitles, which shows post-editing MT result is on average 40% faster than translating from scratch [56]. However, a general evaluation about bilingual subtitle production seems undone yet.

2.3 Sentence Boundary Detection

2.3.1 Multi-Modal Structure Analysis

The structure of multi-modal SBD solutions can be different and has been discussed before [57]. Some researchers propose a single hybrid model, which simultaneously takes all possible features, no matter lexical or acoustic, together as the model input [42, 58, 59, 60], as shown in Figure 2.1-a. Some others apply a structure of sequential models, in which the output of model A is fed into model B, where model A accepts either lexical or acoustic features only, while model B takes the other group of features together with model A's output [40, 41, 45, 61, 62]. This situation is shown in Figure 2.1-b.

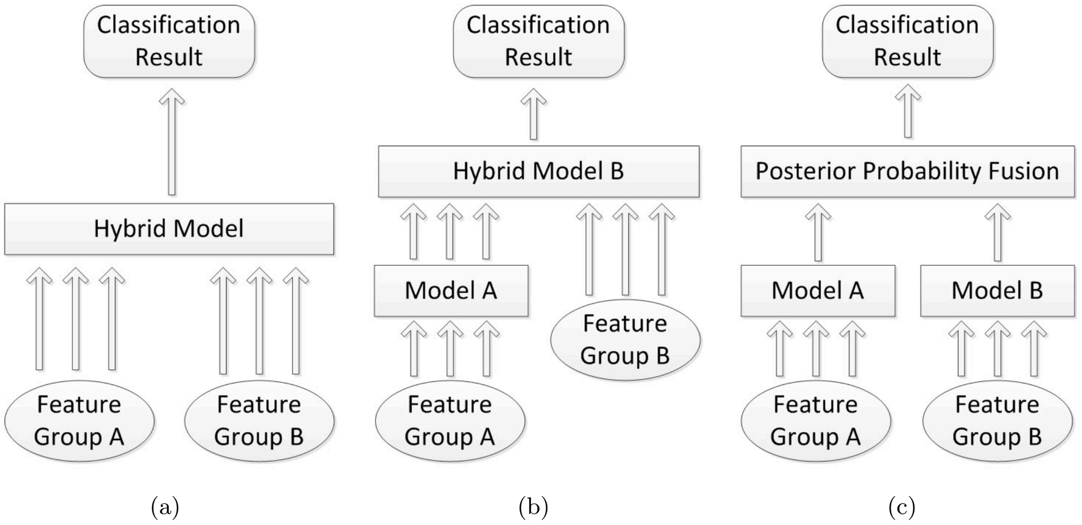


Figure 2.1: Three types of multi-modal SBD structures.

The single model structure is the most straightforward design, but it has one serious shortcoming: the training data must be word-level synchronized transcripts and audios. Obviously, the textual data, which contain lexical features, are far more than audio transcripts and almost inexhaustible. Such structure excludes probably 99% of lexical training data, while the scale of training data is considered vitally important for the classification performance of such models.

The sequential training structure could partially reduce the negative impacts brought by unbalanced training data. Model A in Figure 2.1-b can be first pre-trained as independent model by extra training data, but output high-level features instead of classification result. However, the bottleneck of word-level synchronized transcripts still exist when training model B. Especially when ASR transcript is used for training, the inevitable ASR errors, some of which may be acoustically understandable but lexically ridiculous, such as misrecognizing "*how can we* ***find information*** *in the web*" as "*how can we* ***fight formation*** *in the web*", will definitely downgrade the functionality of the lexical side.

Fortunately, a structure of parallel models can overcome above limitations. As shown in Figure 2.1-c, models can be trained separately with unrelated data. It means the lexical model can take the inexhaustible and grammatically error-free textual data, such as books, newspapers, Wikipedia, unsynchronized manual transcripts of audios, *etc.*, in training. For the acoustic model, all available training data of previous two structures are still available. The next step is to fuse their posterior probabilities.

Gotoh *et al.* and Liu *et al.* trained models with different feature sources separately and interpolated their posterior probabilities for the final prediction [63, 64]. Lee and Glass applied log-linear model to combine outputs from different models [43], so did Cho *et al.* [46]. Pappu *et al.*, on the other hand, used logistic regression model for the fusion [65]. All these approaches offer predictions in one step and need an activation dataset to adjust fusion model parameters.

However, the proposed SBD solution applies a heuristic 2-stage scheme. Different from some earlier multi-pass attempts [38, 44], which first predict punctuation positions and then distinguish punctuation mark types, the two stages in proposed decision scheme is like segmenting and sub-segmenting. Once the independent lexical or acoustic model in this structure is updated, it can be freely switched without re-training or re-activating. The details about this scheme will be introduced later in Chapter 2.3.4, after the descriptions of the lexical and acoustic models proposed.

2.3.2 Lexical Model

2.3.2.1 Technical Plan

As already discussed in Chapter 2.2, commonly used lexical features in SBD are nothing more than LM scores, POS-tags, *etc.* However, a group of such man-made scores or tags are only a collection of some attributes of the corresponding word, which can hardly represent its semantic meaning. If a word could be represented in a computer-understandable way, which means some kind of mathematical structure and in meantime, the semantic meaning of the word could be somehow preserved, it would be great not only for SBD, but also for most of ***N****atural* ***L****anguage* ***P****rocessing* (NLP) tasks. Luckily, word vector provides such a possibility.

W*ord* ***V****ector* (WV), or addressed as word embedding, is developed based on the early concept of distributed representation, which was first proposed by Hinton [66]. The modern word vector derives from the training process of neural language models by Bengio *et al.* [67]. However, WV was not the original purpose of such neural language model training, only a by-product. But it didn't take long for researchers to unveil its value. By simply exporting the word vectors, Collobert *et al.* reports the gains in various NLP tasks "from scratch" [68].

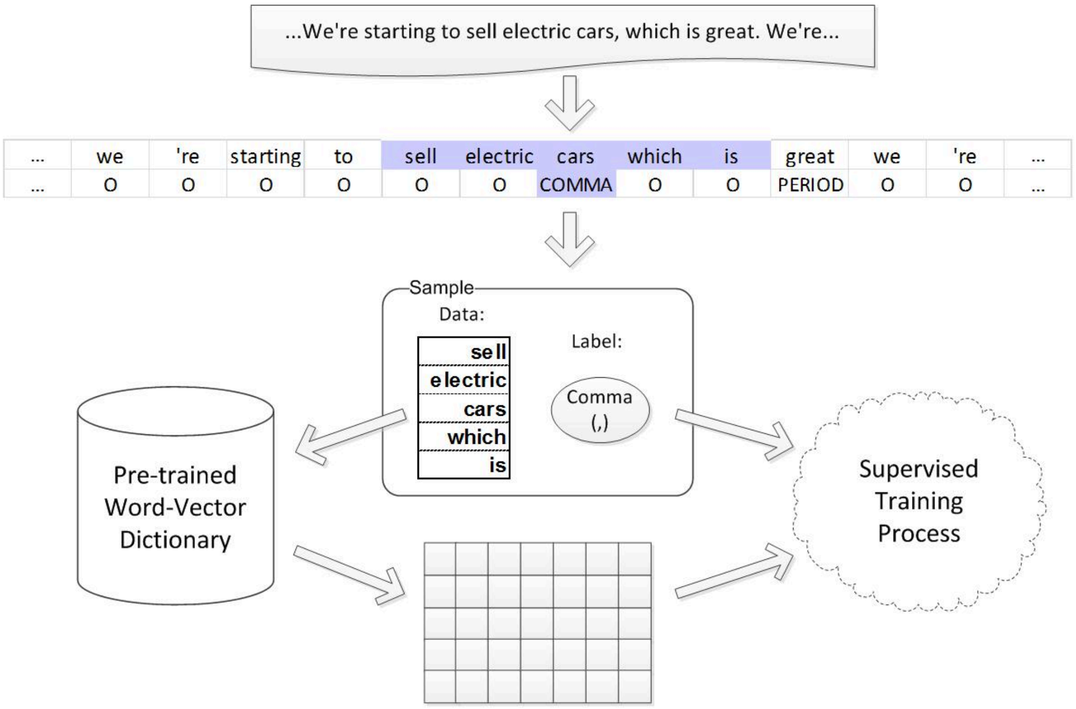
A word vector is in fact a real-valued vector, which is much lower dimensional when comparing with the traditional one-hot representation of words. A more commendable phenomenon is that the semantic distance between words can be measured by the mathematical distance of corresponding word vectors [69, 70]. A frequently mentioned example is: $V_{King} - V_{Man} + V_{Woman} \approx V_{Queen}$. More characteristics about word vectors can be found in Chapter 2.3.5, where an attempt of WV training will be introduced.

Addressing SBD, Cho *et al.* has included word vector in punctuation prediction task together with many other lexical features [46]. However, since the solo usage of word vectors has already been proven effective "from scratch", in proposed lexical SBD models, word vector will be taken as the only input feature, which could also significantly simplify the process of data preparation.

Another issue to consider in technical plan is to select the machine learning model for training. In recently years, ***D**eep **L**earning* (DL) has achieved tremendous success in many researching domains and is especially good at processing raw input data [71], just like word vectors. Several typical DL architectures, such as ***D**eep **N**eural **N**etwork* (DNN), ***Conv**olutional Neural **Net**work* (ConvNet), ***R**ecurrent **N**eural **N**etwork* (RNN) and one important variant of RNN – ***L**ong-**S**hort **T**erm **M**emory* (LSTM), are all widely applied and highly successful.

In NLP domain, the combination of WV and DNN or ConvNet has achieved remarkable performance in sentence classification or sentiment analysis [72, 73], while a typical success with LSTM is sequence-to-sequence translation [74]. Theoretically, ConvNet explores more on local connections [75] and LSTM is better with long context [76]. DNN, on the other hand, is structurally much simpler, which makes the computational complexity much lower when comparing with ConvNet or LSTM on similar scale.

Considering all above factors, the technical plan of proposed lexical SBD model will be "WV+DNN" or "WV+ConvNet". A sliding window with fixed


...We're starting to sell electric cars, which is great. We're...

...	we	're	starting	to	sell	electric	cars	which	is	great	we	're	...
...	O	O	O	O	O	O	COMMA	O	O	PERIOD	O	O	...

Figure 2.2: The process of input data preparation for lexical SBD model training.

length will be applied to create samples, rather than a continuous sequential input. To be noticed, if there is no special explanation, the model will work with English data.

2.3.2.2 Input Preparation

In this approach, the training data are extracted from punctuated textual files, which will be first transformed into a long word sequence with a parallel sequence of punctuation marks. Then an m-words slide window will traverse the word sequence to create samples. The classification question is whether there is a sentence boundary after the k-th word in the sample and if so, which type of punctuation mark it should be.

Four classes are defined in total: O (*means not a boundary*), Comma, Period and Question. All other punctuation marks will be switched into one of above based on their functionalities, such as exclamation marks and semicolons are considered as period, while colons and dashes are classified as comma. Some others, like quotation marks, will be just ignored.

Then each word in the sample will be represented by an n-dimensional word vector which is stored in a pre-trained dictionary. A default vector is prepared as the substitute of any word out of the vocabulary. When processing English data, the substitute word is “this”, because most of the out-of-vocabulary words are proper nouns which exist in special context only, such as “ConvNet” mentioned before, and “***ConvNet*** *is applied in ...*” is grammatically acceptable when switching into “***This*** *is applied in ...*” in purpose of SBD. As a result, an $m \times n$ feature matrix is obtained as the lexical model input for the sample, as shown in Figure 2.2. The training process is supervised by the labels and the value of all WVs are kept static.

2.3.2.3 Model Candidates

In order to pursue optimized performance, three model candidates are proposed and will be tested in preliminary experiments. These three models are addressed as *DNN-3l*, *ConvNet-1d* and *ConvNet-2d* respectively.

DNN-3l is a typical deep neural network with 3 hidden fully connected layers. The $m \times n$ input feature matrix will be reshaped into a 1-dimensional vector, which equals to concatenate the m word vectors one by one. This vector acts as the input layer of the *DNN-3l*. Then training sample can be represented as $\{X_i, Y_i\}$, $i = 1, 2, ..., N$, where X_i is the i-th sample in the training dataset while Y_i is the corresponding label. Then X_i and Y_i can be further represented as $X_i \in \mathbb{R}^{m\times n}$, $X_i = \{x_1, x_2, ..., x_{m\times n}\}$, while $Y_i \in \mathbb{R}^K$, $Y_i = \{y_1, y_2, ..., y_K\}$. The goal of training is to find the optimal weights W and biases b within cost function C:

$$\arg\min_{W,b} C = \frac{1}{2N}\sum_{i=1}^{N} \|\hat{Y}_i - Y_i\|^2 + \frac{\lambda}{2}\sum_{l=1}^{L-1} \|W_l\|_F^2 \tag{2.1}$$

where $\hat{Y}_i$ is the predicted output, while the second term in (2.1) is the weight decay. It is implemented to prevent overfitting, with the weight decay parameter λ which is set to 0.0005 in our case. Sigmoid function as (2.2) is applied to all hidden layers and softmax function as (2.3) to the output layer.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

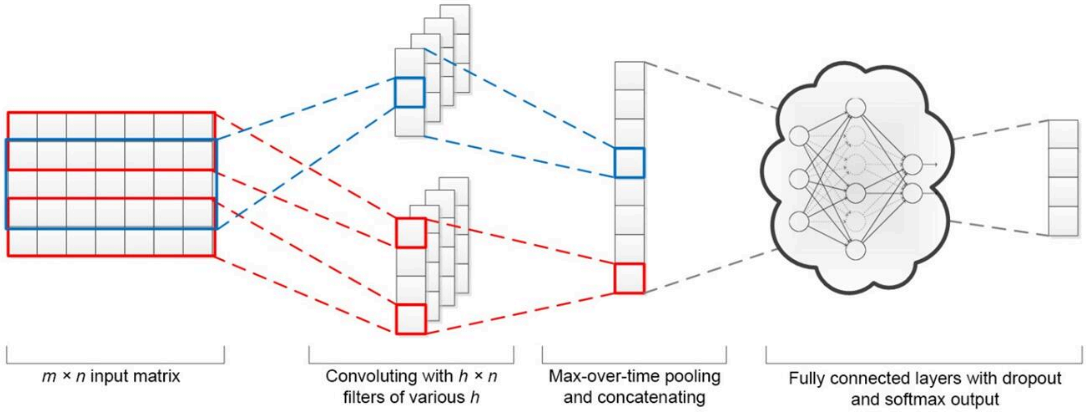

$m \times n$ input matrix
Convoluting with $h \times n$ filters of various h
Max-over-time pooling and concatenating
Fully connected layers with dropout and softmax output

Figure 2.3: Model architecture of *ConvNet-1d*.

$$S_k(y) = \frac{e^{y_k}}{\sum_{j=1}^{K} e^{y_j}}, k = 1, 2, \dots, K \tag{2.3}$$

In precaution of co-adaptation between neurons in the hidden layers, “dropout” function is employed into fully connected layers, which randomly “hide” some neurons along with their connections to reduce overfitting [77]. Dropout is also applied in the fully connected layers in other two models.

The architecture of *ConvNet-1d* is illustrated in Figure 2.3. Taking $m \times n$ matrix as input, a convolutional filter with the kernel size of $h \times n$ will be applied, in which $1 \leq h < m$. Because the value of n is fixed, equal to the dimension of word vectors, the convolutional filter actually can only move vertically. That is the reason why it is addressed as *ConvNet-1d*. Every time the filter captures and processes h continuous words within the sample. After traversing the sample, a feature map with $m - h + 1$ elements will be generated, with each element referring to a convolution position. Then a max-over-time pooling operation will extract the maximum value in this feature map as the final feature of this filter.

By implementing a group of such convolutional filters, with same or different value of h, multiple features can be achieved and then concatenated together, which will be further fed into two fully connected layers with a softmax output. Technically, there is only one convolution layer, but a parallel one. In *ConvNet-1d*, the integrity of word representation is guaranteed during the process of convolution. This model structure is partially inspired by [72].

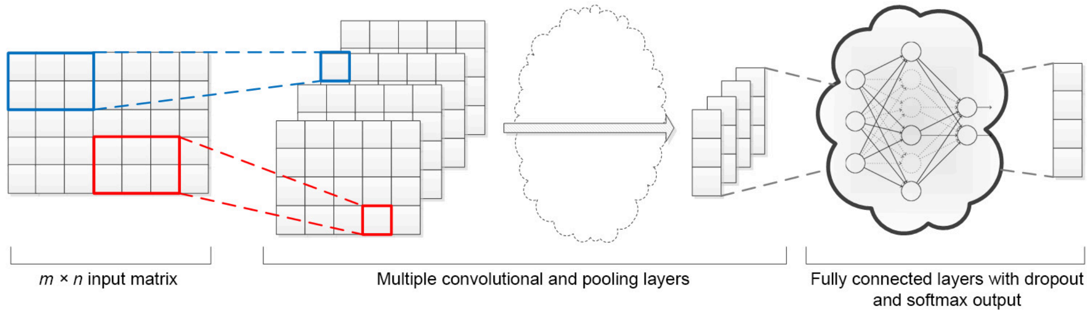

$m \times n$ input matrix
Multiple convolutional and pooling layers
Fully connected layers with dropout and softmax output

Figure 2.4: Model architecture of *ConvNet-2d*.

ConvNet-2d is a more general convolutional neural network. The input $m \times n$ feature matrix is no longer treated as m integrated word vectors. Just like an image with $m \times n$ pixels, each element in the matrix will be treated independently. The width of convolutional filters is not restricted to n, so the filter could move in 2 directions. There are 3 sequential convolution layers and 1 pooling layer in *ConvNet-2d*, followed also by 2 fully connected layers and softmax output. Figure 2.4 illustrates this architecture.

2.3.2.4 Preliminary Experiments and Formal Implementation

As introduced in input preparation, the model would attempt to predict whether there is a sentence boundary after the k-th word in an m-words sliding window. So in preliminary experiments, one prior task is to confirm the sliding window parameters: m and k. For this purpose, several small-scale experiments (pe1∼pe3) with different datasets are made, each of which is executed with different combinations of "m-k". The word vectors used in these preliminary experiments is GloVe.6B.50d set[1].

Figure 2.5 shows the performances of different window settings. Here the accuracies reported are not the original classification result, but the accuracies after excluding the true negative samples (*class 'O'*). More details about the evaluation metric will be introduced in Chapter 2.3.6. Although the performances fluctuate a little bit due to the small data scale, it is still not difficult to figure out from Figure 2.5, that no matter with 5-words or 8-words window, when the

[1] https://nlp.stanford.edu/projects/glove/

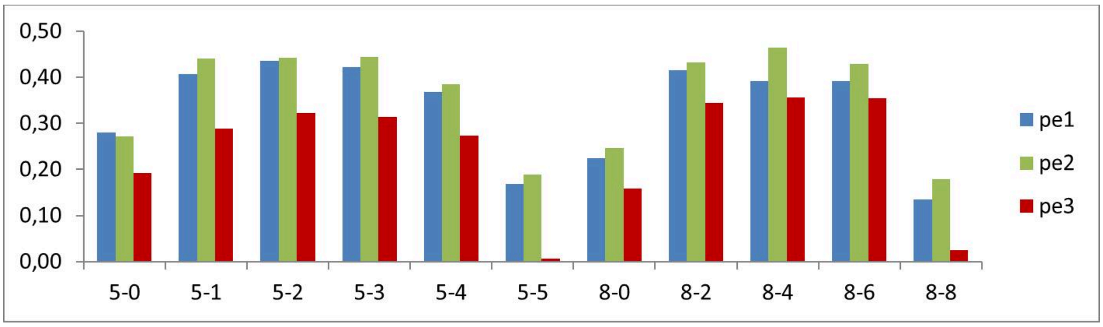


Figure 2.5: The performances of preliminary experiments with different window parameters, for example, '5-3' means a 5-words sliding window is applied while the prediction position is after the 3rd word.

prediction position is closer to the center, the performance tends to be better. This finding will be applied in following experiments.

Another task in preliminary experiments is to select the model from 3 proposed model candidates. In order to fairly evaluate their performances, small-scale datasets are no longer good enough. Therefore a set of formal datasets will be used in model selection, including training set, development set and 2 different test sets. More details about the datasets can be found in Chapter 2.3.6.

In *DNN-3l* model, the numbers of neurons in 3 hidden layers are set to 2048, 4096 and 2048. In *ConvNet-1d* model, 128 "convolution + pooling" filter pairs for each window size from 1 word to 4 words respectively, and two following fully connected layers have 4096 and 2048 neurons accordingly. In *ConvNet-2d* model, the numbers of filters of 3 sequential convolutional layers, with a pooling layer after the first one, are set to 64, 128 and 128, followed by the same fully connected layers of *ConvNet-1d*. "5-3" sliding window and GloVe.6B.50d WV set are applied, and all three models are constructed by CAFFE framework [78].

Table 2.1 shows the performances of 3 model candidates. 4-Classes evaluation means comma, period and question marks are distinguished, while 2-Classes evaluation only cares about whether the position is punctuated. It is clear that *DNN-3l* and *ConvNet-2d* are generally in same level, while *ConvNet-1d* is no match to either of them. However, *ConvNet-2d* took up about 3 times more memories than *DNN-3l* during the training process, as well as the training time is also around 4 times longer. Considering all these facts, it is logical to select

Table 2.1: The performances of preliminary experiments with different model candidates.

	Model	4-Classes			2-Classes		
		Prec.	Rec.	F_1	Prec.	Rec.	F_1
Ref	DNN-3l	**60.3**	48.6	**53.8**	**85.8**	69.2	76.6
	ConvNet-1d	55.2	46.4	50.4	83.0	69.8	75.8
	ConvNet-2d	57.8	**49.9**	53.5	83.6	**72.3**	**77.5**
ASR	DNN-3l	**54.4**	45.6	**49.6**	**77.5**	64.9	70.7
	ConvNet-1d	49.9	45.2	47.5	73.9	66.9	70.2
	ConvNet-2d	51.0	**46.8**	48.8	74.2	**68.1**	**71.0**

Table 2.2: Configurations of lexical SBD implementation in formal experiments. ("Voc.", "n" , "m" and "k" represent vocabulary, vector dimension, sliding window size and supposed boundary position respectively)

Config.	Word Vector			Sample Size	
	Source	Voc.	n	m	k
LMC-1	GloVe	400k	50	5	3
LMC-2	Word2Vec	3M	300	8	4
LMC-2+	Word2Vec	3M	300	10	5

DNN-3l as the final training model in lexical SBD task.

In formal implementation of the lexical model, three configurations will be offered. The first one is the same as *DNN-3l* introduced above, using GloVe.6B.50d word vectors and "5-3" sliding window, but renamed into LMC-1 (***L**exical **M**odel **C**onfiguration*). The second one, addressed as LMC-2, is more informative by applying Word2Vec-Google-300d word vectors[1] and a larger "8-4" sliding window. The third one is similar to LMC-2, just further extending the sliding window size to "10-5" (*addressed as LMC-2+*). Detailed parameters of these configurations can be found in Table 2.2. More experiments and analyses will be discussed in Chapter 2.3.6.

[1] https://code.google.com/archive/p/word2vec/

2.3.3 Acoustic Model

In this subsection, two acoustic models are proposed for discussion. Different from the 4-classes lexical model, acoustic model outputs only 2 classes: boundary or not. The first acoustic model proposed here is a simplest heuristic model. For a word W_i in ASR transcript, the model simply calculates the pause duration p between W_i and W_{i+1} and uses a variant of Sigmoid function:

$$P_a = \frac{1 - e^{-4p}}{1 + e^{-4p}}, \quad p \in [0, +\infty) \tag{2.4}$$

to project p into P_a, while $P_a \in [0, 1)$. Approximately when the pause is longer than 0.28 second, it will be acknowledged as a sentence boundary by this simplest model, which we would like to address as “Pause” in later evaluation process. Since many ASR tools offer timing information per word in their output, this pause-only model could possibly work without the audio file.

The second acoustic model takes more features into consideration and must be applied on the audio file. Pitch level and energy level are extracted by Aubio[1] and Yaafe[2] toolkits. Then based on the timing information in ASR transcripts, an average pitch level or energy level can be achieved for each recognized word. Similar to LMC-2 in lexical model, an “8-4” sliding window is applied. Therefore, each sample contains 25 features: pitch and energy values for the 8 words and 9 available pauses between and around them. These features will also be fed into a neural network with 3 fully-connected layers for training. Based on the feature involved, this second model is addressed as “PPE” (***P**ause, **P**itch & **E**nergy*).

2.3.4 Joint Decision Scheme

As already mentioned in Chapter 2.3.1, the relation between the two stages of the proposed joint decision scheme is like “segmenting” and “sub-segmenting”. Stage 1 takes the posterior probabilities of both lexical and acoustic models as input and detects the sentence boundary positions (*1st Round ‘Hard’ Boundary*). After that the initial long word sequence can be split by these detected positions into segments. Then each segment will be further checked by the adjusted lexical

[1] https://aubio.org/

[2] http://yaafe.sourceforge.net/

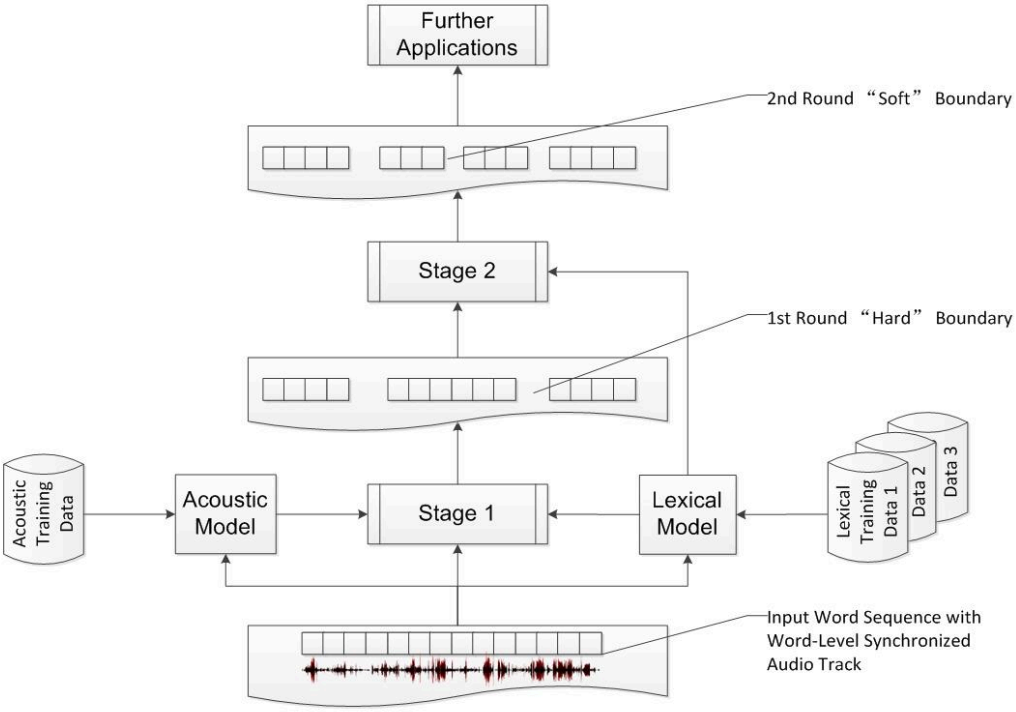


Figure 2.6: The workflow of proposed Joint Decision Scheme.

model output for potential sub-segmenting in Stage 2, generating “2nd Round ‘Soft’ Boundaries”. Figure 2.6 illustrates this procedure.

In Stage 1, acoustic model output is the foundation. Ideally the sentence boundaries in the speech should always result in something detectable in acoustic features, especially the pauses. But actually the speaker may hesitate or be interrupted by unexpected events. These phenomena cause false positive detections in acoustic analysis, so the lexical model is employed here as a “filter”. The basic idea is that if an acoustically supposed boundary position is strongly opposed lexically, it will be denied. The lexical denial threshold is associated with the confidence of acoustic prediction by a simple linear function. If P_a and P_l are used to represent the posterior probability of “being a boundary” from acoustic and lexical model respective, then a hard boundary will be confirmed only when $1 - P_l < P_a \times 0.25 + 0.7$ and $P_a > 0.05$.

In Stage 2, only lexical model output is considered. The goal is to recover the sentence boundaries which have no acoustic hint. Since many boundaries have already been detected in stage 1, only the positions with strong lexical evidence will be classified as soft boundaries, for which the posterior probability of lexical model is adjusted by

$$P_l' = P_l \times e^{(\frac{L}{\hat{L}}-\lambda)} \times \frac{d \times (L-d)}{(\frac{L}{2})^2} \tag{2.5}$$

where L is the length of the input segment, d is the distance between current word and previous detected boundary, $\hat{L}$ is the expected length between adjacent boundaries and λ is the restriction coefficient. This adjustment generally reduces the P_l. In practice the value of $\hat{L}$ and λ are fixed and the extent of reducing becomes smaller when L gets larger and d approaches $L/2$, which means a soft boundary is supposed to be found in the middle position of a long input segment. In extreme case, the adjustment might even increase P_l, but L needs to be more than λ times larger than $\hat{L}$, which may hardly happen. Generally, only positions with very strong lexical evidence to be boundaries can be acknowledged after the adjustment as (2.5).

Basically the joint decision scheme works with only 2 classes: boundary or not. But when punctuation marks are required to be restored, it can be fulfilled in Stage 2 based on the lexical model in use. Suppose n types of punctuation marks are available, then $P_l = \sum_{i=1}^{n} P_i$. As long as a position has already been confirmed as boundary, no matter a hard boundary or soft boundary, the i-th type of punctuation mark will be chosen when P_i is the largest in $\{P_1, P_2, ..., P_n\}$.

2.3.5 Extension to Other Languages

2.3.5.1 German WV Training

On openHPI and tele-TASK platforms, there are also quite a lot of e-lectures instructed in German. Therefore it is practically meaningful to extend subtitling service from English to German, including SBD. But not like in English, it is difficult to find reputable public WV set in German, thus training our own WVs is necessary.

As already mentioned in Chapter 2.3.2.1, modern WVs derive from neural language model, as the exported coefficients of the input-projection layer or so-called “look-up table” [67]. Mikolov *et al.* adjusted the network into a RNN to specialize WV training and published the tool "*Word2Vec*", which provides two working schemes: CBOW and Skip-gram. Simply speaking, CBOW uses the context of several neighboring words to predict current word, while Skip-gram uses the current word to predict its context [69]. The theory behind such context-based training is the distributional hypothesis proposed by Harris [79], who claimed that "*a word is characterized by the company it keeps*".

Some later efforts are made to further improve this "company", such as introducing corpus-based statistics (*GloVe*) [70] or defining neighbor words based on parsing dependency tree instead of writing order (*word2vecf*) [80]. In order to solve the problem of out-of-vocabulary words in morphologically rich languages, Bojanowski *et al.* proposes "*FastText*", which trains vectors of sub-word strings along with words: when a out-of-vocabulary word appears, it can be represented by the sum of the vectors of those sub-word strings contained [81].

However, for the consistency with English approach, original *Word2Vec* is chosen for the German WV training. Data are collected from Leipzig corpus [82], Wikipedia dump and some domain-specific text. But firstly a small portion of collected data is adopted for preliminary experiments, in order to optimize the training parameters. These preliminary results show that higher dimension of WV and larger window size for context could always benefit in semantic features capturing, but not so significant with syntactic features, while unavoidably increase the training complexity. The performances of CBOW and Skip-gram models are similar.

Considering all these findings, the parameters are finally set to: dimension = 300, window size = 10, epoch = 10 and Skip-gram is using.

2.3.5.2 German WV Evaluation

In order to evaluate the quality of WVs achieved, a typical point of view is to see whether the semantic distance between words is measurable by the mathematical distance between corresponding WVs. A frequently used evaluation task is to find the nearest word of the given word in vector space, and then compare with human judgement. But a more vivid manifestation of word similarity is to reduce the dimensionality of WVs and project them in a visualizable 3D or 2D space. Ideally, words with similar meanings or functionalities should locate closely.

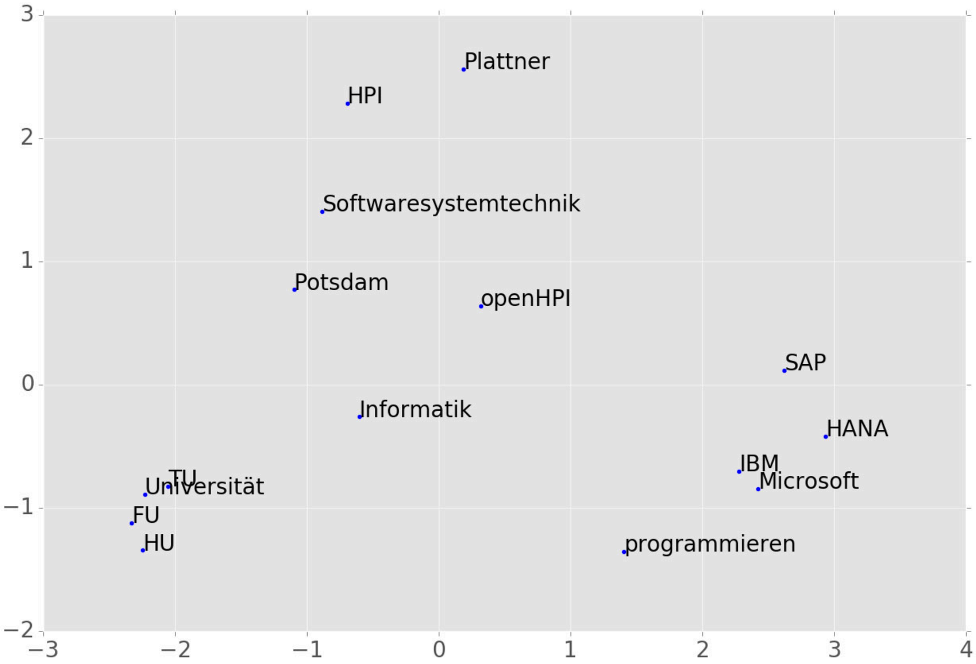


Figure 2.7: The visualization of selected WVs in a 2D space.

Figure 2.7 gives and example from the German WVs trained, in which a group of selected words are projected on a 2D space. As can be seen, the words addressing universities, including "Universität", "TU", "FU" and "HU" (***T**echnische, **F**reie and **H**umboldt **U**niversität, all of them are universities in Berlin*), are closely gathered together, while IT companies like "IBM", "Microsoft" and "SAP" are also locating not far from each other. HPI as an independent IT institute, on the other hand, is neither a university nor a company, and its location is actually far from these two groups. The closer neighbors of HPI include "Plattner" (*its founder*), "Softwaresystemtechnik" (*its focus*), "Potsdam" (*its location*) and "openHPI" (*its program*). Basically, the distribution of words involved in Figure 2.7 makes sense.

Another popular evaluation task of WVs is word analogy question. Just as the example introduced in Chapter 2.3.2.1, $V_{King} - V_{Man} + V_{Woman} \approx V_{Queen}$, the word analogy questions are of the form "A is to B as C is to D", where D must be predicted by calculation. With proper questions, both the semantic relations and the syntactic correctness can be measured. It is also difficult to find public

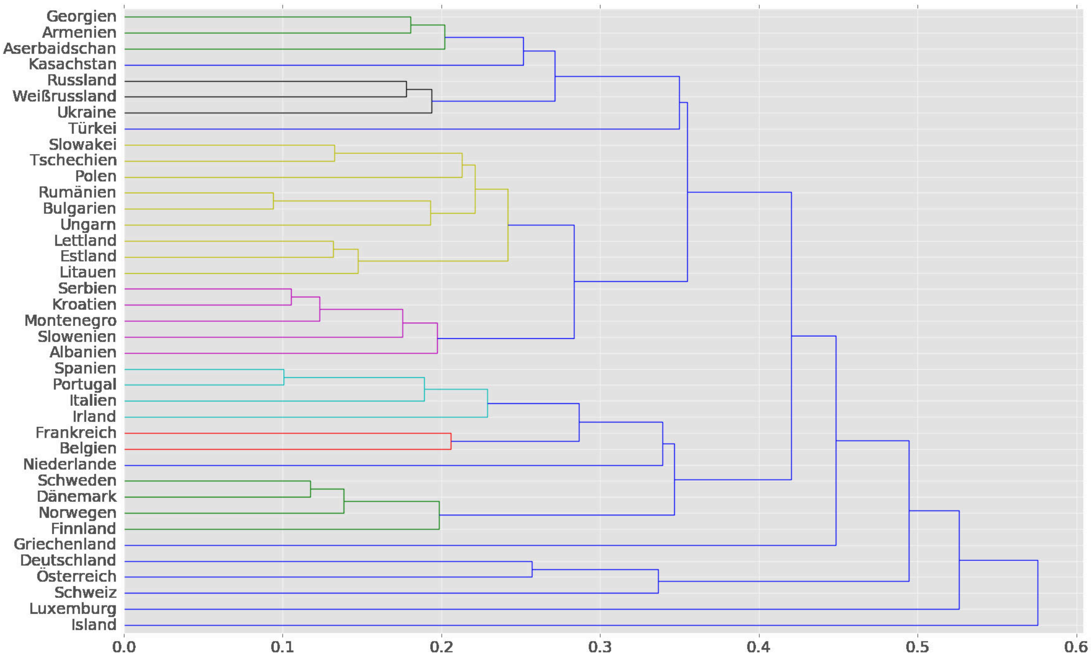


Figure 2.8: The clustering of WVs related to some countries.

word analogy questions in German, so 2834 semantic questions in 18 categories and 77886 syntactic questions in 9 categories are defined (*can be found online*[1]). The German WVs trained can reach 38.6% accuracy in semantic questions and 50.1% in syntactic questions.

Besides, the WVs are also evaluated by a proposed alternative metric of word analogy test. Instead of predicting the fourth word by the given three, the new metric directly measures the similarity on the “relations” of two pairs of given words, just as shifting the relation vectors into a new analogy space of same dimensionality. In this way, word analogy questions can be measured in a more comprehensive way by also including the reverse word logic, while avoiding the traversal of the whole vocabulary and thus saving the evaluation time significantly by over 95%. With this metric, German WVs trained return similar stats with reputable public English WVs.

A more intuitive display of the characteristics of German WVs achieved can be seen in Figure 2.8. It is a progressive clustering task of (mainly) european

[1] https://drive.google.com/open?id=0B13Cc1a7ebTuaE83NEtyemM4aGM

Table 2.3: Configurations of lexical SBD implementation in German

Config.	Word Vector			Sample Size	
	Source	Voc.	n	m	k
d-LMC-2	W2V-HPI	1.1M	300	8	4
d-LMC-2+	W2V-HPI	1.1M	300	10	5

countries. Once two closest WVs are found and clustered together, the average value will be applied as the representation of this cluster. Apparently, the first round clusters are highly logical by geography (*Transcaucasia, Scandinavia, Baltic, Iberia, etc.*), history (*ex-Yogoslavia, ex-Czechoslovakia*) or language (*German-speaking, French-speaking*). Even in later rounds, the clustering is still reasonable, like connecting eastern european countries or Catholic countries together. Based on Figure 2.8, it is fair to say that in the semantic of european nations, the German WVs trained are highly capable.

Unfortunately, it is impossible to thoroughly evaluate the quality of WVs with above tasks. So the best evaluation is to apply them in actual use, and in this case, in SBD.

2.3.5.3 SBD Implementation in German

With pre-trained German WVs, same lexical model which works with English data could also work with German data, since the word segmentation method in the written version of these two languages is the same. The WVs are trained with Word2Vec toolkit, which has the same format of Word2Vec-Google-300d English WV set used in English, so the configurations in German SBD are also set similar to the configurations in English SBD: d-LMC-2 and d-LMC-2+. They are prepared with sliding window sizes of “8-4” and “10-5” respectively. More details can be seen in Table 2.3.

2.3.5.4 Potential Further Extensions

Theoretically, word vectors can be trained for almost any languages and further activate corresponding lexical SBD service. Just as the extension from English to German, exactly same processing method can be directly applied on almost all

European languages based on letters and using spaces to segment adjacent words, such as French, Italian, Russian, Greek, *etc.*, with same or different alphabets.

However, more language-specific efforts need to be made when dealing with languages with other systems, such as Chinese. The basic unit in written Chinese is character, which may represent a monosyllabic word or a morpheme, but definitely functions more than a letter. Since there is no space between adjacent semantic words in written Chinese, many NLP tasks have to take word segmentation as a beginning step [83]. WV training and SBD can also adopt this solution.

But there are also some discussions about whether it is necessary to introduce Character Vector for Chinese language instead of WV [84, 85], or at least including the characters as sub-word units in WV training [86]. The previously mentioned "*Fasttext*" may perfectly fit the condition of Chinese, since the structure of "characters in word" is definitely more logical than arbitrary string of continuous letters in word. Nonetheless, with limited adjustment, SBD service can be extended to Chinese. It should be also feasible for other languages with proper knowledge and efforts.

2.3.6 Evaluation

2.3.6.1 Datasets

The datasets used in SBD evaluation are collected originally from IWSLT datasets. For the lexical model, performances with both ASR and manual transcripts are evaluated (*for English*). The test set is the "tst2011" package for IWSLT 2012 ASR Track, which consists of 8 TED talks and has both ASR and manual transcripts, containing around 12k words each. The training data consists of the manual transcripts of 1710 TED Talks and comes originally from the in-domain training data of IWSLT 2012 MT Track. The data are further split into training set and development set, with 2.1M and 296k words respectively. There is no overlapping between datasets, checking by the talk IDs. These datasets are made publicly available[1].

For the acoustic model, the range of data selecting is quite limited. The 8 TED talks contained in "tst2011" are still taken as the test set, while 70 other talks

[1] https://drive.google.com/open?id=0B13Cc1a7ebTuMElFWGlYcUlVZ0k

with synchronized ASR transcripts and audio files collected from different IWSLT datasets make up the training set. There is no development set for acoustic model evaluation. Joint solution will also be tested on these data. Additionally, a special small lexical training dataset is prepared with the transcripts of the 70 TED talks used in acoustic training, which contains approximately 80k instances in total, in order to figure out how the lexical model can perform with limited training data.

For German SBD, only lexical model will be evaluated. German datasets are built based on IWSLT 2015 MT Track, with only manual transcripts available. 1597 talks are then split into training and development sets, with 2.8M and 310k instances respectively. The test set consists of the transcripts of another 16 talks, with 22k words in total.

2.3.6.2 Metrics

The average length between two adjacent boundaries in the English training set is 7.8, which means over 85% of the instances belong to "O" class, and the majority of these "O" instances can be successfully classified. However, excessive "true negative" instances would make the classification accuracy so high, that the influences of other "punctuated" classes in SBD task will drastically decline and become more difficult to meature. Therefore, in all evaluation tasks in this chapter, correctly classified "O" instances will be excluded and the performances will be evaluated by:

$$Precision = \frac{\#\ Correctly\ predicted\ punctuation\ marks}{\#\ All\ predicted\ punctuation\ marks} \tag{2.6}$$

$$Recall = \frac{\#\ Correctly\ predicted\ punctuation\ marks}{\#\ All\ expected\ punctuation\ marks} \tag{2.7}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.8}$$

As briefly mentioned in preliminary experiments in Chapter 2.3.2, evaluation on lexical models has two aspects: 4-Classes evaluation which distinguishes punctuation mark types and 2-Classes evaluation which only focuses on sentence boundary positions. Besides, the detailed statistics about comma, period and question mark will be presented in a "per-class" evaluation. On the other hand,

Table 2.4: Performances of lexical models (*in percentage*)

Test Set	Model	Tr-Size	4-Classes			2-Classes		
			Prec.	Rec.	F_1	Prec.	Rec.	F_1
ASR-EN	CRF-Best	unknown	47.8	54.8	51.0	–	–	64.0
	T-LSTM	2.1M	49.1	43.7	46.2	69.3	61.6	65.2
	LMC-1	2.1M	54.4	45.6	49.6	77.5	64.9	70.7
	LMC-2	2.1M	54.0	52.2	**53.1**	76.8	74.2	**75.5**
	LMC-2-80k	80k	45.6	23.5	31.0	77.8	40.1	52.9
Ref-EN	CRF-Best	unknown	49.8	58.0	53.5	–	–	75.8
	T-LSTM	2.1M	55.0	47.3	50.8	75.3	64.6	69.5
	LMC-1	2.1M	60.3	48.6	53.8	85.8	69.2	76.6
	LMC-2	2.1M	60.4	55.8	**58.0**	85.8	79.3	**82.4**
	LMC-2+	2.1M	60.3	54.5	57.3	85.2	77.1	81.0
Ref-DE	d-LMC-2	2.8M	64.3	64.3	64.3	79.0	79.0	**79.0**
	d-LMC-2+	2.8M	64.8	64.3	**64.5**	79.1	78.5	78.8

performances of acoustic models and joint decision scheme are only reported by 2-Classes evaluation.

2.3.6.3 Lexical Model Evaluation

In this subsection, the lexical model would be tested on both ASR transcripts and manual references of English TED talks, which are addressed as "ASR-EN" and "Ref-EN", while German SBD performance would be reported in "Ref-DE". On both English test sets, the performances of LMC-1, LMC-2 along with two baseline approaches for comparison, CRF-Best [35] and T-LSTM [62], are reported. Please note that the dataset used by CRF-Best is not exactly the same as what introduced in Chapter 2.3.6, although it was also claimed to be "tst2011" from IWSLT. Additionally, the performance of LMC-2 trained by limited 80k words would also be tested on ASR-EN, while on Ref-EN, LMC-2+ is tested. Within Ref-DE, both d-LMC-2 and d-LMC-2+ are experimented. The general accuracies can be found in Table 2.4.

It is clear that LMC-2 is the best performer in both ASR-EN and Ref-EN tasks, while LMC-1 is generally in the same level of CRF-Best and outperforms

T-LSTM. However, it should not neglected that the *DNN-3l* model used by LMC-1 and LMC-2 contain several thousands of hidden neurons, which is apparently complicated than T-LSTM model described in [62]. Since CRF-Best and T-LSTM were the best performing models which could be found and compared with by the time when LMC-1 and LMC-2 were published, it is fair to say that the proposed model has reached the state-of-the-art in purpose of lexical SBD. And it is quite understandable that LMC-2 functions better than a less informative LMC-1. Moreover, the performance of LMC-2-80k on ASR-EN is way worse than LMC-2, which clearly shows the crucial importance of sufficient training data in a lexical SBD approach.

In Ref-DE tests, the German model d-LMC-2 achieves better 4-Classes but lower 2-Classes accuracies with its corresponding English model LMC-2, which means in English LMC-2, there are many commas are recognized as periods, vice versa, but this phenomenon happens much less frequently in German d-LMC-2. However, the general outcome of German SBD is in same level of English SBD. This result suggests several findings:

1. The quality of German word vectors trained in Chapter 2.3.5 is proven to be good with practical application. If confined to the discussion of SBD task, the German WVs is functionally approaching the highly reputable public "Word2Vec-Google-300d" English WV set.
2. The proposed lexical SBD model is applicable and effective to multiple languages, as long as there are corresponding pre-trained word vectors.
3. The opposite accuracies of 4-Classes and 2-Classes show the inherent linguistic difference between English and German, which would imply the potential of language specific solutions or adjustments in SBD.

Another issue in lexical model evaluation is about LMC-2+ and d-LMC-2+. By simply extending the sliding window from "8-4" to "10-5", it is not improving in German and even getting worse in English. This fact may suggest that based on the sliding window sampling scheme, perhaps "8-4" configuration is already near the roof. If a breakthrough is desired, some more advanced and complicated structure should be considered. More discussion will be presented in association with "per-class" evaluation.

Table 2.5: Performances of lexical models per class (*in percentage*)

	Model	COMMA			PERIOD			QUESTION		
		Pre.	Rec.	F_1	Pre.	Rec.	F_1	Pre.	Rec.	F_1
ASR-EN	T-LSTM	41.8	37.8	39.7	56.4	49.4	52.7	55.6	42.9	**48.4**
	LMC-1	47.2	32.0	38.1	59.0	60.9	60.0	–	0.0	–
	LMC-2	45.1	46.5	**45.8**	63.4	60.1	**61.7**	–	0.0	–
Ref-EN	T-LSTM	49.6	41.4	45.1	60.2	53.5	56.7	57.1	43.5	**49.4**
	LMC-1	58.2	35.7	44.2	61.6	64.8	63.2	–	0.0	–
	LMC-2	55.1	51.0	**53.0**	65.6	63.5	**64.6**	66.7	5.0	9.3
	LMC-2+	53.6	48.8	51.1	66.8	62.4	64.5	75.0	19.6	31.0
Ref-DE	d-LMC-2	63.9	62.9	63.4	65.2	72.0	**68.4**	56.7	12.4	20.4
	d-LMC-2+	64.6	63.4	**64.0**	65.1	70.9	67.9	58.8	14.6	**23.4**

Table 2.5 shows the "per-class" performances of several models. All models are better in detecting periods than commas, regardless of datasets or languages. From the author's point of view, it is quite logical, since the ambiguity of a pause is generally higher than a full-stop, especially in less formal text like the transcripts of TED talks. Generally, the German models are much better with commas than English, probably because the grammar system of German language is stricter, which is also a common sense.

The performance of question marks detection is also interesting. LMC-1, with a "5-3" sliding window, fails in classifying even one question mark, while LMC-2 and d-LMC-2 with extended "8-4" sliding window are not much better either: the recall rate is never beyond 20%. The author believes there are two reasons behind this awkwardness. The first is the proportion of "question" samples in the training data is much lower than others, resulting in insufficient training. The second, which might be more crucial, is that no matter the "5-3" or "8-4" sliding window could hardly cover the corresponding sentence beginning.

By comparing the "5-0" & "5-5", "5-1" & "5-4" or "8-2" & "8-6" pairs in Figure 2.5 in preliminary experiments, it may suggest that in lexical SBD, the context after the prediction position is more important than the context before. In other words, it is more likely the beginning of a new sentence or clause to be detected, rather than the end of the current one. However, this theory may only work with commas and periods, not the quantitatively disadvantaged question marks. A typical question pattern, such as "*what do you ...*" or "*how can I ...*", locates only at the beginning of the sentence.

If the above analysis makes a point, it could explain why the window size extension from "8-4" to "10-5" does not improve the general performance, but apparently increases the accuracy with question mark, because an extra word involved in previous context enlarges the possibility to involve the key question word "what" or "how". It may also explain why the referenced T-LSTM method works only better with question marks than proposed methods, because dealing with long previous context is exactly the major advantage of LSTM model.

Considering all technical conditions, analyses and inferences, a bi-directional LSTM could be a worthy attempt in purpose of lexical SBD in future work. Currently, LMC-1, LMC-2 and LMC-2-80k will be further offered for the joint decision scheme evaluation, along with proposed acoustic models.

2.3.6.4 Acoustic Model and Joint Solution Evaluation

Table 2.6 shows the results of proposed acoustic model together with joint solution. The test set is the same as ASR-EN in lexical evaluation. The performances of acoustic models, "Pause" and "PPE", can be found in "Acoustic" column of Table 2.6. Since there are only 2 classes available for the acoustic models, corresponding lexical posterior probabilities are also referred in 2-Classes only. Based on the lexical and acoustic models proposed, the testing results of 6 possible combinations are presented in two phases: "Joint-S1" shows the result after the decision scheme Stage 1, and "Joint-S2" is the final result.

The performances of two acoustic models are almost the same, both of which are higher than LMC-2-80k, but lower than LMC-1 and LMC-2. The results of all combinations after Stage 1 are around 10% better than acoustic-only, but when best lexical performer LMC-2 is adopted, the "Joint-S1" result cannot compete with the lexical-only performance yet. However, it is reasonable, since Stage 1 is designed to filter false positive acoustic detections only. And it actually results in a comparatively high precision but low recall rate, just as expected. Then Stage 2 does improve the recall rate a lot. In the end, the performance of a joint solution is better than either lexical or acoustic model.

Table 2.6: Acoustic model and joint solution evaluation (*in percentage*)

Models	Lexical	Acoustic	Joint-S1			Joint-S2		
	(F_1)	(F_1)	Prec.	Rec.	F_1	Prec.	Rec.	F_1
LMC-1 + Pause	70.7	60.9	82.8	62.3	71.1	79.2	76.0	77.6
LMC-2 + Pause	75.5	60.9	85.5	62.1	71.9	78.8	79.5	**79.2**
LMC-2-80k + Pause	52.9	60.9	83.5	58.0	68.5	79.2	65.5	71.7
LMC-1 + PPE	70.7	61.0	77.1	67.4	72.0	75.8	76.6	76.2
LMC-2 + PPE	75.5	61.0	79.7	67.5	73.1	76.5	80.6	**78.5**
LMC-2-80k + PPE	52.9	61.0	78.4	62.1	69.3	75.9	66.4	70.8

In many previous works, researchers claimed that pause is the dominant acoustic feature in SBD task [62, 63, 87, 88]. It is also what this evaluation shows. The "Pause" model works as good as "PPE" model independently, although "PPE" model consumes much more calculating resources. When working jointly with lexical model, the combinations with a simpler "Pause" model manage to achieve even better result.

In proposed parallel model structure, the lexical model is more important. With fixed acoustic model, the combination with better lexical model always achieves better result. However, the differences between the performances become smaller after fusing the lexical probabilities with the acoustic model output. Finally, the best performer in SBD, "LMC-2 + Pause", will be applied in the process of automatic subtitle production.

2.4 Automatic Subtitle Production

2.4.1 Framework

When producing subtitles manually, the procedure can be generally divided into three steps: transcription, timeline alignment or synchronization, and translation. These steps could be done by one person (*staff-A*) and then proofread by others (*staff-B*). The framework of proposed automatic subtitle production solution is in similar structure, as seen in Figure 2.9, which consists of ASR, SBD and MT. Due to technical limitations, a manual modification process by staff-A might be needed in practice before proofreading by staff-B.

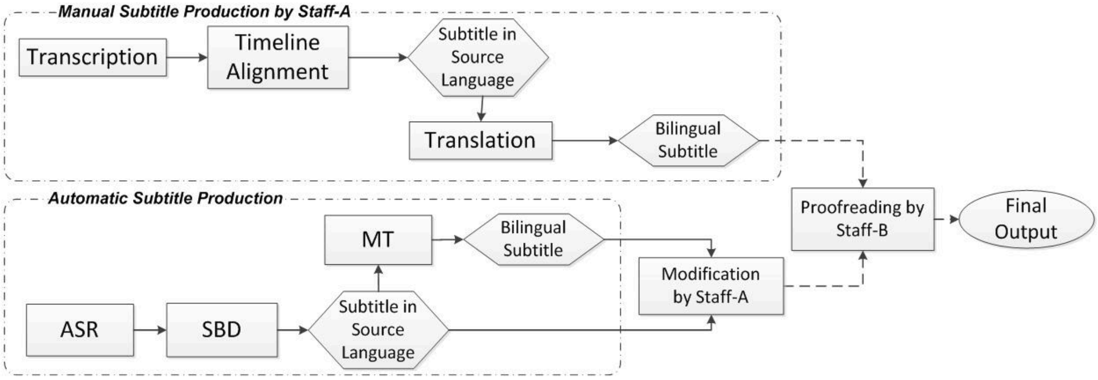


Figure 2.9: The procedure of subtitle creation: manual & automatic.

In proposed solution, SBD is the major technical contribution and has been explained in detail previously. In this section, a special configuration to SBD, which is oriented to the characteristics of subtitle, will be introduced. For both ASR and MT, 3rd-party services are implemented. However, these procedures would still be introduced by the order in Figure 2.9.

2.4.2 ASR

IBM Watson Speech-to-Text service is chosen as the ASR tool [89]. By submitting the audio file to the ASR server through API, the transcript file in .json format could be retrieved within 1.5∼2 times of the audio duration. The transcript contains timestamps for each word and has also been segmented into sentence units. However, this default segmentation is far from good enough, so it will be removed before being fed into the proposed SBD tool.

Since IBM Watson Speech-to-Text does not offer German ASR service yet, Google Speech API is used as the substitute when actually dealing with German lectures. By simply adjusting the format of ASR output, every ASR tool could be implemented. The choice of IBM Watson is made based on the balance of ASR quality and pricing.

2.4.3 SBD with Special Configuration

As introduced in Chapter 2.3.6, the SBD tool could process the ASR output with an approximate accuracy of 80%. If the purpose is to offer lecture transcript in paragraph, it could be ended after restoring possible punctuation marks in the text. But as supplementary material to the video, subtitle could only be displayed in a limited region inside the video player, commonly in the bottom area, otherwise it may cover the major visual content [48]. Therefore, one single subtitle item should have a maximum length, up to two rows in general principle [47, 48]. And in bilingual subtitles, each language could only occupy one row.

Based on some tests with the lecture videos of openHPI, the maximum length per item is set to 60 Latin characters when processing English or German. When a grammatically correct sentence or clause, or more plainly, a continuous word sequence between two adjacent restored punctuation marks is longer than 60 characters, which is in fact happening frequently, especially with German lectures, further segmentation is needed.

In such cases, the lexical SBD model will be reactivated to find a most possible re-segmenting position. This process runs recursively until all segments are shorter than the maximum length. By this approach, basic grammar units could be avoided from splitting. For instance, the line break will never take place between "take" and "place" or after an article "an". There will be no punctuation marks added in these line break positions.

Besides, a minimum length of 15 Latin characters is also set. Lecture videos, especially those prepared for MOOCs, generally belong to solo-speaking video, which are probably shot in a professional studio instead of a real classroom [90, 91]. In this situation, the lecturer could keep a calm and stable speaking manner and there is hardly any "yes or no" question answering scenario. Thus if a subtitle item is too short, its duration is very likely to be short either. When displaying, these short subtitle items may appear as "flashing", which does no good for the learners' watching experiences [47]. So the short items will be combined with corresponding following items and the punctuation marks will be kept when exist. If the new item is consequently over the maximum length, a line break will be found in the previous "following item".

Now with proper formatting, the production of subtitle file in source language could be concluded. Since the SBD model is pre-trained, the SBD process of a 10-minutes lecture can be completed in less than a minute. The time expenditure is approaching zero.

2.4.4 MT

The MT tool adopted is Microsoft Translator API. The textual content of the previously generated subtitles in source language will be submitted to the translation server item by item. The returning text in target language will be directly added as a second line in the corresponding subtitle item. In this way, bilingual subtitle production is accomplished. Roughly one second is needed for the translation of every two subtitle item. For the convenience of further manual modification and proofreading, subtitle file with only target language will not be provided from the automated process.

2.5 Subtitle Quality Evaluation

2.5.1 Methodology

In order to evaluate the quality of auto-generated subtitles and their contribution in process of e-lecture preparation, accuracy is undoubtedly important. However, as already mentioned in Chapter 2.1, since auto-generated subtitles inevitably contain errors despite of the significant development of ASR and MT, it is perhaps inappropriate to offer them directly to online learners. Therefore a more important function of auto-generated subtitles is serving as the draft to facilitate manual post-editing. So in this section, a quantitative evaluation about how these auto-generated subtitles can help will also be included.

As illustrated in Figure 2.9, manual subtitle production consists of transcription, timeline alignment and translation. Comparing with the steps in automatic framework, the tasks of transcription and timeline alignment are actually covered by “ASR + SBD”, with standardized subtitle file in source language as output. MT further adds content in target language. So it is logical to evaluate the subtitle quality in source language and target language separately.

A few short lecture video clips selected from openHPI learning platform are taken as the testing data, with English as teaching language. Then 24 volunteers are invited to help producing subtitles for comparison. The volunteers are fluent but non-native English speakers, who come from China, Germany, Russia, Iran and Indonesia respectively. There are two reasons for this arrangement: the first is that based on past experiences, subtitle production is generally taken charge by staffs who speak the target language, not the source language; the second is that it was impossible to find enough native English speakers at that time. In experiments, all volunteers works independently. Ground-truth subtitles are created by several experts, including the corresponding lecturer himself.

For source language evaluation, each volunteer would receive two video clips, V_1 and V_2, and one auto-generated subtitle file, which might be suitable for either V_1 or V_2, addressed as A_1 or A_2. If a volunteer received A_1, he/she is expected to create the modified subtitle S'_1 for V_1 based on A_1 and the manual subtitle S_2 for V_2 from scratch, vice versa. Meanwhile, the volunteers are demanded to report the time expenditures on individual tasks. In this way, all volunteers are divided into two groups, working on "S'_1+S_2" or "S_1+S'_2" separately. Any volunteer would not work with same video twice, by which the performance deviation caused by the volunteer's different familiarities with certain video can be avoided. Since all volunteers have worked with both videos, the difference of English skills between them can also be balanced. In the end, the average error rate of A_i, S_i and S'_i, $i \in \{1, 2\}$, can be evaluated in comparison with ground-truth, while the average time expenditure of S_i and S'_i can also be calculated in a fairly convincing way.

For target language evaluation, only English-to-Chinese translation is tested. After excluding non-Chinese speakers from the volunteers, others are also divided into two groups with different translation tasks. Each volunteer in group 1 would receive two video clips, one error-free English-only transcript E_1 and one machine-translated bilingual subtitle B_2. He/she would be expected to create Chinese subtitle C_1 by translating A_1 and modify B_2 into C'_2, as well as the time expenditures. Similarly, volunteers in group 2 should finish "C'_1+C_2". After collecting all possible C_i and C'_i, average accuracy and time expenditure are measured.

2.5.2 Evaluation with Source Language

The video clips used in source language evaluation derive from the welcome video of openHPI platform. V_1 starts at 0:29 and V_2 starts at 2:02, both of which last 64 seconds. They are intentionally kept short to avoid bringing too much burden to the volunteers. In many NLP tasks, such as ASR and MT, ***Word***

Table 2.7: Evaluation with source language

	Auto-Generated (w/ or w/o SBD)						From Scratch			Post-Edited		
	A_1		A_2		$\bar{A}$		S_1	S_2	$\bar{S}$	S'_1	S'_2	$\bar{S}'$
	w/o	w/	w/o	w/	w/o	w/						
T	–	–	–	–	–	–	1001	845	**923**	489	360	**424**
P1	.147	.143	.087	.070	.117	**.107**	.154	.117	**.135**	.081	.043	**.062**
P2	.178	.186	.120	.096	.149	**.141**	.184	.143	**.164**	.116	.074	**.095**

__E__rror __R__ate (WER) is widely used as the measurement of accuracy. However, it is probably improper to take WER in such short fragments, since the total number of words is very limited. Alternatively, character-level Levenshtein Distance [92] is applied, which may also be addressed as edit distance. Generally, the Levenshtein Distance is the minimum number of single-character edits needed, including insertion, deletion and substitution, to change one string into another.

Accuracy evaluation consists of two phases. Phase 1 focuses on textual content only, which ignores the subtitle item boundaries by connecting all items in a single long string, with one space between every two adjacent subtitle items. Phase 2 takes subtitle item boundary positions into consideration, with each boundary counted as a 3-character word. By taking ground-truth subtitle file as anchor, a Levenshtein Distance between the under-test subtitle file and the anchor can be calculated. The ratio of this Levenshtein Distance and the under-test file length is further acknowledged as error rate, which will be addressed as "P-1" and "P-2" for the two phases. Time expenditure (T) is counted in seconds.

The accuracies of auto-generated subtitles are reported in two configurations, with (w/) or without (w/o) SBD. With "w/" setting, SBD is executed and punctuation marks are restored, which is also what offered to the volunteers. With "w/o" setting, on the other hand, default ASR segmentation is applied and the transcript is unpunctuated. From the statistics in Table 2.7, it is easy to figure out that the error rate of auto-generated subtitles is around 10∼15%, which is fairly acceptable.

However, a more encouraging finding is the significant improvement in subtitle quality and the unneglectable achievement in time saving when taking the auto-generated subtitles as draft in post-editing. The pure textual error rate drops

Table 2.8: Evaluation with target language

	Machine-Translated			Human-Translated			Post-Edited		
	B_1	B_2	$\bar{B}$	C_1	C_2	$\bar{C}$	C'_1	C'_2	$\bar{C}'$
Time	–	–	–	636	853	**744.5**	675	671	**673**
Total Items	20	17	18.5	20	17	18.5	20	17	18.5
Modifications	10	9	9.5	2.1	3.9	3	4.9	3.6	4.3
Error Rate	0.500	0.529	0.514	0.106	0.228	**0.162**	0.244	0.213	**0.230**

from 0.135 to 0.062 (54.3%), while the ASR baseline (0.107) is in the middle. In boundary included evaluation, the error rate also drops round 42.1%. Meanwhile, the working time can be shortened by 54% averagely. More discussion could be found later in “Result Analysis”.

2.5.3 Evaluation with Target Language

The video clips used in target language evaluation are selected from lesson 6.3 of the MOOC “Web Technologies” in 2015, talking about search engines. V_1 starts at 0:05 and V_2 starts at 5:22, with the same duration: 103 seconds. Different from source language evaluation, there is no ground-truth for translation task in this context. The accuracy will be evaluated manually with following principles:

- ◇ The actual translators of the MOOC “Web Technologies” are invited as reviewers.
- ◇ Subtitle items are taken as the units in evaluation.
- ◇ Each unit would be evaluated on whether modification is necessary, based on its meaning, grammar and fluency.
- ◇ The error rate is defined as how many units need to be modified.

The statistics of target language evaluation can be found in Table 2.8, in which time expenditure is also counted in seconds. Different from what achieved in source language, the contribution of MT in subtitle production is less convincing. Especially for V_1, the quality of post-edited Chinese subtitle is relatively worse than direct manual translation from English. On the other hand, working time saved is also limited, around only 9.6%.

2.5.4 Result Analysis

Firstly, automatic subtitle production in source language has been proven to be very helpful. The quality of the auto-generated subtitle is already better than the average of manual subtitling from scratch, which offers a good starting point for post-editing. By taking it as draft, human subtitle producer could achieve higher accuracy with less time spent.

Another contribution of auto-generated subtitle is that there is no need for human producers to care about timeline alignment issue. In manual producing, the theoretical minimum time expenditure for the timeline alignment is the same as the video duration. However, even with professional subtitle processing software, it is still very difficult to avoid misoperation. So consuming more time on this task is quite normal, especially for unexperienced staff.

In the proposed framework, SBD is the technical focus and has been elaborated in detail. The error rate of the subtitle generated with SBD is indeed lower than those without SBD, as shown in Table 2.7. Besides, around 1/3 of the volunteers keep the detected sentence boundaries, or the line breaks in corresponding subtitle files unchanged in post-editing, which is probably one reason of the shortened working time. Since the translation tasks in our experiments are based on error-free transcripts, the influence of SBD in MT unfortunately cannot be measured.

It is not ideal that the auto-generated subtitle in target language is way less helpful to human producers. One major reason might be that the quality of machine translated text is far from good enough. It is clear in Table 2.8 that the error rate of baseline is 0.516, which is way higher than the error rate in source language. After struggling with the technical or semantic errors which may mislead the learners, it is somehow understandable for the post-editors to be over tolerant upon those technically or semantically not wrong, but lexically unnatural or disfluent translations, which could be further marked as “modification needed” by reviewers. Particularly, the difficulty of translation between English and Chinese is widely acknowledged.

Finally, the total time expenditure throughout the complete procedure of both manual production from scratch and automatic generation with post-editing will be estimated, step by step according to Figure 2.9, before proofreading. Suppose the duration of the input lecture video is d, based on the statistics in Table 2.7 and the corresponding 64 seconds testing clips, manual transcription time can be calculated as $14.4d$. Similarly with Table 2.8, the manual translation time is $7.2d$. In addition with $1.5d$ for timeline alignment, the total expenditure is $23.1d$. If automatic subtitling is applied, with $2d$ for ASR, $0.2d$ for SBD and MT, $6.6d$ for source language post-editing and $6.5d$ in target language modification, the total expenditure of time is $15.3d$. Roughly, over 1/3 of the total working time can be saved, while keeping the output in similar or even better quality.

2.6 Chapter Summary

In this chapter, an integrated framework of automatic bilingual subtitle production is presented, which consists of ***A**utomatic **S**peech **R**ecognition* (ASR), ***S**entence **B**oundary **D**etection* (SBD) and ***M**achine **T**ranslation* (MT). SBD is the major technical contribution among them. The quality of the subtitle generated is evaluated not only by accuracy, but also by a quantitative volunteer-based research on how it can help with the online course preparation.

The proposed SBD approach applies a structure of parallel lexical and acoustic models. The lexical model is built on ***D**eep **N**eural **N**etwork* (DNN) and takes ***W**ord **V**ector* (WV) as the only input feature. Evaluation shows the proposed lexical model reaches the state-of-the-art performance. Along with a simple but effective pause-only acoustic model, the posterior probabilities of parallel models are fused by a 2-stage joint decision scheme, which further improves the SBD accuracy.

In order to extend SBD service from English to German, a set of German WVs is trained based on Word2Vec toolkit. According to the linguistic characteristics of German, some adjustments are made to both the tool and the training data. A German-specific evaluation shows that quite a lot of syntactic and semantic connections between words are captured by the WVs achieved, and the implementation to SBD also obtains very encouraging result.

By sequentially executing ASR, SBD and MT, a bilingual subtitle file can be created. The accuracy of auto-generated subtitle is fairly high, while a quantitative evaluation with 24 volunteers involved shows that by taking the auto-generated subtitle as draft, human producer can roughly save 1/3 of the total working time with no quality drop. Especially with the source language, over 50% gain in both accuracy and time expenditure can be achieved simultaneously.

With the subtitles generated, e-lecture service can definitely be enhanced for non-native online learners. The proposed solution has already been used in preparation of multiple MOOCs and is highly appreciated by the teaching teams.

Chapter 3

Lecture Outline Generation

3.1 Motivation

Suppose such a scenario: a learner sits in front of a computer screen and searches for a lecture about a certain topic he/she wants to learn. However, there are so many options returned under the searching terms the learner just typed in. With very limited further information, the learner is unable to make a better choice than randomly clicking on one. After carefully watching the video for several minutes, the learner closes it because it is not exactly what he/she wants ......

It is very frustrating and something should be done to help with it. As already mentioned in Chapter 1, e-learning providers should offer proper descriptions about the video-based lectures to facilitate learners in selection. By far the most frequently used method to do so is tagging, which has been researched over decades. But no matter the tags come from user feedback [93, 94, 95], or derive by automated semantic analysis [96, 97], the information carried by tags is very limited. It might be enough to address "what it is", but not "how it goes", let alone to offer an adequate lecture preview. Manual description is another possibility, which could contain enough information to work as an abstract. However, it is impossible to hire a group of professionals or force the lecturers to create such descriptions, which takes too much time and/or money.

Lecture outline might be a better option. Some studies suggest that students benefit from the lecture outline when taking online courses [98, 99]. A survey offered with MOOC "Internetworking" on openHPI.cn shows that 91% of the

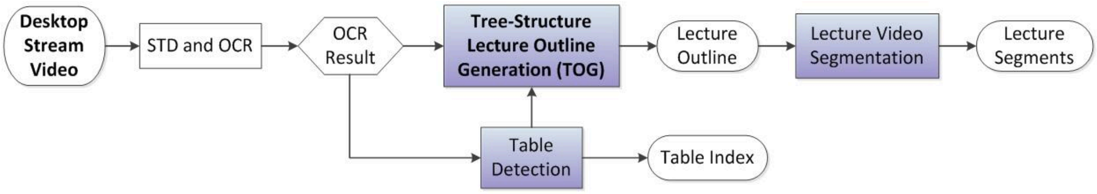


Figure 3.1: The basic structure of proposed technical solutions in Chapter 3.

respondents (*90 of 99*) believe an accurate outline could be a positive factor in their learning process. A proper outline contains much more information that tags and is much better structured than description, which enables multiple functions like preview, navigation and retrieval.

In this chapter, an automated solution of lecture outline generation is proposed. The outline is in fact extracted from lecture slides which are synchronized with lecture videos. Slides have occupied the front of the classroom in recently years [100, 101] and are included in many online courses. And more importantly, people use slides as the outline of the talk, so do the lecturers who prepare the lectures. With double-stream lecture video recording systems, such as tele-TASK, a desktop stream is generally used to record the real-time slides displaying. This desktop stream video is taken as the starting point of the proposed solution.

Figure. 3.1 shows the basic framework of proposed solution. ***S****lide* ***T****ransition* ***D****etection* (STD) and ***O****ptical* ***C****haracter* ***R****ecognition* (OCR) are applied on the desktop stream video, to first obtain screenshotted slide images and then extract textual data from them. The OCR accuracy can reach 92% on character-level and 85% on word-level [102]. Based on the OCR result, ***T****ree-Structure* ***O****utline* ***G****eneration* (TOG) is developed, which is also the major contribution in this chapter. Table detection is originally one step in TOG, but due to its independence and extensibility, it would be introduced as a sub-application of TOG. Besides, a video segmentation approach is also implemented after achieving the lecture outline. It would be taken as a follow-up application.

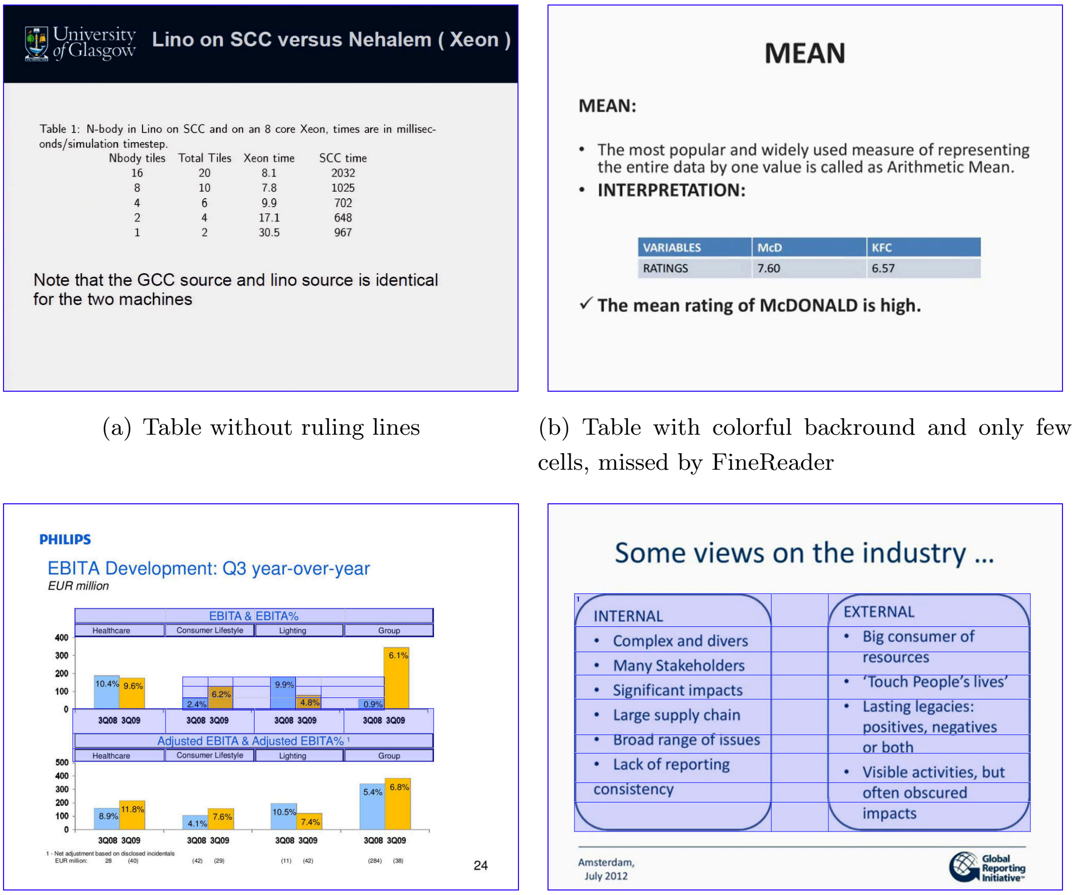
The rest of this chapter is arranged as follow: all related works would be discussed in Chapter 3.2. Table detection, as a sub-application and necessary step of outline generation, will be first introduced in Chapter 3.3. Chapter 3.4 explains the outline generation process in detail and Chapter 3.5 illustrates the outline-based lecture video segmentation application. A chapter summary can be found at the end.

3.2 Related Works

The research works about lecture outline creation involve different techniques. ASR is one option, by which a lecture outline is supposed to be concluded from the lecture transcript based on some NLP methods [103, 104]. However, even if ASR is assumed to be 100% correct and the highly successful deep learning technique is applied, text summarization is still a highly challenging task [105, 106]. Capturing the teacher's writing on the blackboard is another option [107, 108]. But since the usage of blackboard in classroom is decreasing and the blackboard is not always included in online lectures, these approaches are less practical.

When using lecture slides as the data source, Atapattu *et al.* proposed a solution to explore the hierarchical semantic concepts from the slides and further saved in tags [109]. Li and Dong claimed that typical presentation slides can be hierarchically parsed [110]. Yang *et al.* focused on a different aspect by locating important textual components in slides and arranging them in a list to facilitate user's navigation [111]. By combining these inspiring ideas together, it evolves into the simplest version of TOG, focusing on inter-slides and intra-slide phases respectively. But in general, the study in lecture outline is still in preliminary status.

On the other hand, table detection is a long term research topic in document analysis area. Most researchers focus on traditional portrait-oriented, text-dense and book-like documents. If the documents are saved in digital PDF files, metadata extraction is the most effective method to detect tables [112, 113]. When the documents are in form of scanned images, ruling line detection [114, 115] and white space analysis [116, 117] are most popular. In a table detection contest in 2013, commercial software ABBYY FineReader is the best performer, with accuracy over 98% [118]. Meanwhile, the contest organizer reported two factors which cause difficulty: lack of ruling lines and small tables with fewer than five



University of Glasgow

Lino on SCC versus Nehalem (Xeon)

Table 1: N-body in Lino on SCC and on an 8 core Xeon, times are in milliseconds/simulation timestep.

Nbody tiles	Total Tiles	Xeon time	SCC time
16	20	8.1	2032
8	10	7.8	1025
4	6	9.9	702
2	4	17.1	648
1	2	30.5	967

Note that the GCC source and lino source is identical for the two machines

(a) Table without ruling lines

MEAN

MEAN:

- The most popular and widely used measure of representing the entire data by one value is called as Arithmetic Mean.
- INTERPRETATION:

VARIABLES	McD	KFC
RATINGS	7.60	6.57

✓ The mean rating of McDONALD is high.

(b) Table with colorful backround and only few cells, missed by FineReader

(c) False positively detected table from a diagram by FineReader

Some views on the industry …

INTERNAL		EXTERNAL
• Complex and divers		• Big consumer of resources
• Many Stakeholders		• 'Touch People's lives'
• Significant impacts		• Lasting legacies: positives, negatives or both
• Large supply chain		
• Broad range of issues		
• Lack of reporting consistency		• Visible activities, but often obscured impacts

Amsterdam, July 2012

Global Reporting Initiative

(d) False positively detected table from a 2-columns layout slide by FineReader

Figure 3.2: The challenges of detecting table from slides

rows. Unfortunately they are quite common in slides, just as Figure 3.2-a and Figure 3.2-b[1].

The diversity of slide layouts causes other problems, such as dark background with light text, diagrams with lines and annotations, sparse but well-aligned 2-column layout, *etc.* By applying ABBYY FineReader 12 on these example slide images, the table in Figure 3.2-b is missed and false positive detections are made in Figure 3.2-c and Figure 3.2-d. According to these facts, a slide-oriented table detection method is highly desirable, which should abandon all "shortcuts" and

[1] The copyright belong to original slide authors or institutions: (a)Mr. William Cockshott, (b)Mr. Avi Pipada, (c)Royal Philips Electronics, (d)Ms. Tamara Bergkamp

go back to the definition of table: a table is an arrangement of data in rows and columns. In this way, some earlier approaches based on text bounding box clustering [119, 120] could be more referential.

The situation of video segmentation research is similar to table detection. It is also a long term research topic, but in majority for natural videos, with scene change detection as the technical core [121, 122]. However, lecture video is something different, especially lacks of visual scene changes [123, 124]. In order to segment lecture videos, some researchers applied ASR on the lecture video and then attempted to segment the transcript [125, 126]. Some other approaches were proposed as multi-modal, taking both audio and visual signals into consideration [127, 128].

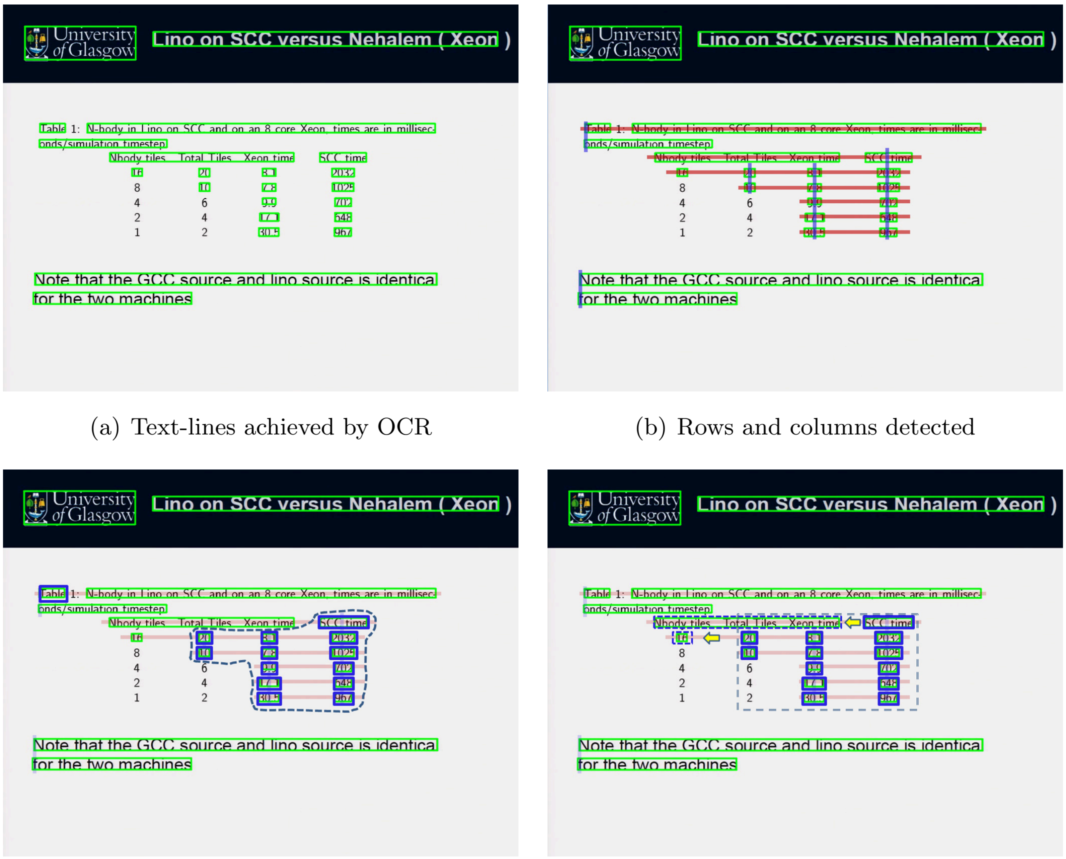
Another possibility is to detect slide transitions. Obviously, the content within a slide is semantically closer than the content distributed in different slides, so it is natural to take slide transition as the clue for lecture video segmentation [129, 130]. Therefore, many efforts have been made to capture slide transition in different situations and enhance the slide images captured [131, 132]. It is also taken as a preliminary step of the proposed TOG. But in purpose of storing and retrieving video segments, directly taking slide transitions as segment boundaries could perhaps make the segmentation result too fragmentary. Some further efforts need to be made.

3.3 Table Detection from Slide Images

3.3.1 Detection of Rows and Columns

Rows and columns are the indispensable elements of a table. Their existence distinguishes a table from other components in a document, such as a paragraph or a diagram. In order to deal with the diversity and ambiguity of table formats in slides, the proposed table detection solution starts with detecting potential structures of rows and columns in the slide images.

As introduced previously, all the textual data within a slide image have been parsed by OCR process and stored in text-lines. A text-line includes both the textual content and the location parameters, based on which a virtual bounding box can be generated, as shown in Figure 3.3-a. Then a slide image can be



University of Glasgow

Lino on SCC versus Nehalem (Xeon)

Table 1: N-body in Lino on SCC and on an 8 core Xeon, times are in milliseconds/simulation timestep

Nbody tiles	Total Tiles	Xeon time	SCC time
16	20	8.1	2032
8	10	7.8	1025
4	6	9.9	702
2	4	17.1	648
1	2	30.5	967

Note that the GCC source and lino source is identical for the two machines

(a) Text-lines achieved by OCR

(b) Rows and columns detected

(c) Intersections and the table candidate

(d) Table area expansion

Figure 3.3: An example of proposed table detection process

considered as a bunch of such bounding boxes and a blank background. Therefore the task is simplified as to judge whether two bounding boxes, or we say, two text-lines locate in a same row or column.

Theoretically it is quite easy to confirm rows. The only requirement is to have two text-lines horizontally locating in a same line. But practically the bounding boxes created for words "Glory" and "name", even with same font, size and actually locating in a same line, may have different heights, and the words "Time" and "map" might even appear interlaced, because of the shapes of letters. In addition with unavoidable and unpredictable OCR errors, a compromised judging mechanism is applied, which requires at least 3/4 of two text-lines vertically overlap and one cannot be twice the height of the other, or more.

Column searching is more complicated and the key issue is the alignment. The cells which belong to same table column could be aligned to the left, to the right or centered. All these three possibilities need to be considered. Additionally, two table cells might even coincidently conform to more than one alignment type, when they have similar widths and horizontal positions. To cope with these situations, following mechanism is applied step by step:

1. List all text-lines within current slide as T_1, T_2,..., T_n, and then traverse all possible text-line pairs $T_{ij} = \{T_i, T_j\}, 0 < i < j \leq n$.
2. If T_i and T_j are not vertically aligned, ignore 3∼6 and go directly to 7.
3. If T_i and T_j are vertically aligned, record all their alignment types in A_{ij}. $A_{ij} \subseteq \{Left, Right, Center\}$ and $A_{ij} \neq \emptyset$.
4. Check whether T_i is already included in any existing column candidate C and whether the intersection of A_C (*the alignment types of* C) and A_{ij} is not empty. ($C \subseteq \{T_1, T_2, ..., T_n\}$, $T_i \in C$ and $A_C \cap A_{ij} \neq \emptyset$)
5. If yes, add T_j into C. And set the intersection of A_C and A_{ij} as new A_C. ($A'_C = A_C \cap A_{ij}$)
6. If no, create a new column candidate $C_{new} = \{T_i, T_j\}$. And set $A_{C_{new}} = A_{ij}$.
7. Continue with next pair.

With above mechanism, it is possible for a column to have more than one alignment type, which in fact has no negative influence for later procedures. It is also possible for a text-line to be shared by multiple columns. In this case, columns with shared cells will be combined together, as long as there are no cells within the new column locating in same row − otherwise no combination will be made and the shared cells will be removed.

This mechanism will create quite a lot of false positive table columns, such as a left-aligned text paragraph, a group of annotation in a diagram, or just several unrelated text-lines coincidently seem to be aligned. Since the elimination of these false positive columns can be gradually applied in later procedures, it is comparably riskier to miss a potential table column in this early stage of the solution. Figure 3.3-b shows all the 7 rows and 5 columns found in the example slide, including false positive ones.

3.3.2 Table Area Positioning

3.3.2.1 Table Candidate Generation

A table cell is supposed to be the intersection of one row and one column. Rows and columns have already been detected and each of them is a set of several horizontally or vertically aligned text-lines. So the intersected text-lines, each of which belongs to both a row set and a column set, are the most likely table cells. They are the foundation to locate the table area.

Since there might be more than one table in one slide, all the intersected text-lines will be grouped by their rows and columns belonged. Any two intersected text-lines which belong to a same row or column will be grouped together and the effect superimposes. At the end, a text-line shares a row or column with at least one other text-line within its group, but does not need to share with all of them. Any 1-member-group will be directly removed, just like the intersection with text “Table” in left-upper corner of Figure 3.3-c. Such intersection groups will be taken as potential table candidates.

3.3.2.2 Table Candidate Evaluation

Here all potential table candidates will be evaluated by four measurements with descending importance. Among them, content mark and standardized column bonus deal with the textual content of the intersected text-lines, while distance deduction and two-column deduction focus on the layout. After accumulating the values of all measurements, only the potential table candidates with positive final value will qualify.

Content Mark measures the likelihood of a text-line to be a table cell based on its content. Strictly speaking, there is no standard or regulation illustrating what can be written in a table and what cannot. However, people prefer to put numbers, percentages, single-words or short phrases into a table, rather than long sentences. So if the content of an intersected text-line belongs to above “table-cell-likely” categories, it would earn a positive value. And as the length of the content increases, the content mark decreases to 0 or negative. After traversing all text-lines in the potential table candidate, an average content mark will be calculated, which could be either positive or negative.

Standardized Column Bonus can be only positive or 0. Commonly a table row is used to address a subject with different attributes, while a column stores the values of a certain attribute from all included subjects, just like previously used Table 2.1∼2.5. When this happens, most cells of a same column contain same type of content, except for the header row. If the content of such a column within the potential table candidate is digit or single-word, it is strong evidence to support the existence of a table and should earn a big positive value in this evaluation. The threshold to determine whether a column contains same type of content is set to 75%, which generally allows one additional exception other than the header row, because actually, it is not rare for the OCR to misrecognize 'O' and '0', or 'I' and '1'. When a table is reversely designed, with columns to represent subjects and rows for attributes, just as Table 2.6 or 2.7, standardized column bonus does not work, which is why it has no negative value.

Distance Deduction functions when two neighboring columns horizontally locate too far away with each other. It is designed for those slides with chart or diagram, whose description texts are sometimes aligned but remotely located. The maximum of 1/8 slide width and the correspondent column's width is taken as the reference. If the gap between two neighboring columns is larger than the reference, a deduction will be applied, and the value of this deduction depends on how much larger the gap is. This measurement can be only negative or 0.

Two-Column Deduction is in precaution of the special two-column slide layout, like Figure 2.2-d, which is a default layout in most of templates of PowerPoint. When this layout is applied, the items in these two columns are very likely to be aligned both horizontally and vertically, which is quite similar to a two-column table candidate and fairly probable to be detected as one by previous procedure. In order to decrease such false positive detections, if a vertical axis can be found in the middle of the slide to make the two columns symmetric, and more than 80% of all text-lines within the whole slide are included in these two columns, a deduction will be applied with a comparatively small negative value.

Detailed settings of above measurements can be found in Appendix A.

3.3.2.3 Table Area Expansion

A table may have some empty cells. Like in a 3 × 3 table, 9 cells should be expected, but it is also possible to have only 6 cells, with 3 in the first row, 2 in the second and 1 in the third, with a shape of triangle. Then it is obviously impossible for the previous procedure to involve the only cell in the third supposed row, because technically this row does not exist. A 2 × 2 table candidate is the optimized result, which is actually only a part of the table. Moreover, OCR error may cause cells missing, resulting in irregular shapes of table candidates, such as the scribbled table area in Figure 3.3-c. In these cases, an expansion process is needed.

Firstly, a virtual rectangle which covers all the text-lines involved in a table candidate is drawn as the initial table area, just like Figure 3.3-d. Then the expansion attempt starts alongside the rows and columns which go across current table area. Once a target text-line is found, its content and location will be checked to decide whether it should be added into the relevant table candidate (*Detailed settings in Appendix A*). If so, the table area will be updated. The expansion process goes in loop until there is no more target can be found.

3.3.3 Table Confirmation

A final confirmation will be made on each table area detected. Since a text block might unlikely "survive" as a false positive table area at this late stage, the main task in the final confirmation is to distinguish those table-like charts or diagrams and eliminate them. One prerequisite is applied before the procedure: if a table area has extreme aspect ratio, such as 10:1 or 1:10, it will also be directly denied, because no actual table should be like that. Different with what has been done in Table Candidate Evaluation, the measurements of the confirmation process focus not only on cells, but also consider the table area as a whole. Quite a lot of factors are considered, including:

- ◇ M_c: the average content mark, supposed to be larger if the table is confirmed.
- ◇ C_T: total cells, about the scale of the table, supposed to be larger.

- ◇ C_E: expected total cells, the product of total rows r and total columns c. The ratio of C_T and C_E is in range of $(0, 1]$ and positively related with the chance of a table area to be confirmed.
- ◇ A_T: the whole table area, counted in pixels, supposed to be larger.
- ◇ H_T: average text height, counted in pixels, supposed to be larger.
- ◇ D: text density, defined as the ratio of the sum of areas covered by all text-lines within the table area and A_T. It should be neither too large nor small, so 0.2 is taken as the benchmark.

Now we need to take all these factors together into a general consideration. Theoretically we know the factors are positively or negatively related with the final result, but on practical level, the distribution of each factor's weight derives from attempts in the training set. The final equation of table confirmation can be illustrated as

$$M_{final} = \frac{e^{M_c}}{3} + \frac{\ln C_T \times C_T}{C_E} + \frac{\ln H_t^2 \times (\ln \ln \sqrt[4]{A_T})^3}{e^{\sqrt[3]{|D-0.2|}}} \tag{3.1}$$

But to be noted, (3.1) has no mathematical theory behind. In later evaluation procedures, the confirming threshold is set to: $M_{final} > 6.5$.

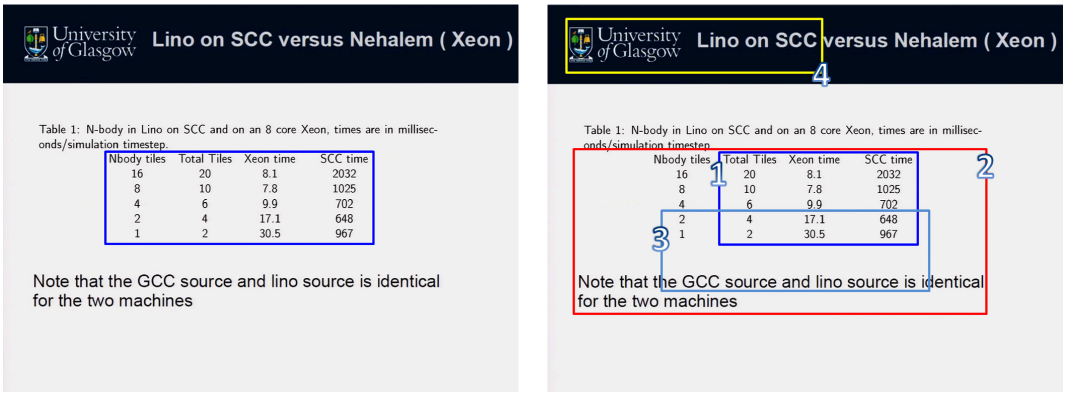
3.3.4 Evaluation

3.3.4.1 Datasets and Metrics

There are three datasets in our evaluation process: training set, benchmark set and test set. Both training set and test set consist of slide images. The training set gathers 493 slides from 12 complete presentations, which contains 46 tables and is collected from tele-TASK.de. The test set includes 384 slides from the excerpts of 26 different presentations on either tele-TASK.de or SlideShare.com[1]. The slides in the test set cover various topics such as economy, education, media, IT, *etc.*, and contain 189 tables in total.

The proposed table detection solution will be compared with open source Tesseract table detection kit [133] and commercial software ABBYY FineReader

[1] https://www.slideshare.net/



Lino on SCC versus Nehalem (Xeon)

Table 1: N-body in Lino on SCC and on an 8 core Xeon, times are in milliseconds/simulation timestep.

Nbody tiles	Total Tiles	Xeon time	SCC time
16	20	8.1	2032
8	10	7.8	1025
4	6	9.9	702
2	4	17.1	648
1	2	30.5	967

Note that the GCC source and lino source is identical for the two machines

Lino on SCC versus Nehalem (Xeon)

Table 1: N-body in Lino on SCC and on an 8 core Xeon, times are in milliseconds/simulation timestep.

Nbody tiles	Total Tiles	Xeon time	SCC time
16	20	8.1	2032
8	10	7.8	1025
4	6	9.9	702
2	4	17.1	648
1	2	30.5	967

Note that the GCC source and lino source is identical for the two machines

(a) Correct detection

(b) Other sorts of detections

Figure 3.4: The categories of possible detection

12[1]. In order to avoid misusing these tools, a benchmark set of traditional documents is also prepared. The benchmark set is shared by the competition organizer of [118] and consists of 238 pages, including 156 tables.

To evaluate the performances, two facts are focused on: how the tables in ground-truth get detected and how accurate an actual detection is. However, a detected table area cannot be simply classified by correct (*Figure 3.4-a*), false positive (*Figure 3.4-b-4*) and missing, because there are more possibilities: partial detection (*Figure 3.4-b-1*), over detection (*Figure 3.4-b-2*) and partial-and-over detection (*Figure 3.4-b-3*). In order to quantify the performances, each detected table will be given a precision weight as 1, 0.75, 0.5, 0.25 or 0, based on the proportion of its accurately detected cells against the ground-truth. Apparently, correct detection values 1, false positive (F.P.) values 0, missing table has no value while the others depend.

By accumulating the precision values of all actual detections, a recall rate can be calculated against the ground-truth (G.T.) and a precision rate against the number of total detections (T.D.). Please note, that any over detected table will also value 1 when calculating the recall, regardless of how much extra area is also involved, because the detection is actually complete. Finally the F_1-Score can be also calculated.

[1] https://www.abbyy.com/en-eu/support/finereader-12/

Table 3.1: Table detection experiments on all datasets. "M.-S." means the method and dataset used, in which P, T & F for the proposed approach, Tesseract and FineReader, while Tr, B, L & H for training set, benchmark set, low-resoluation and high-resolution test set respectively. Recall, precision and F_1 are reported in percentages.

M.-S.	G.T.	T.D.	Detection Categories						Rec.	Prec.	F_1
			Co.	Part.	Over	P&O	Miss	F.P.			
P-Tr	46	39	21	10	5	0	10	3	69.02	73.72	71.29
T-B	156	111	69	16	7	2	61	17	55.61	74.55	63.70
F-B	156	153	126	21	0	0	10	6	88.62	89.87	89.24
T-L	189	107	13	34	12	12	118	36	25.93	41.12	31.80
F-L	189	181	109	41	5	2	35	24	73.81	75.97	74.87
P-L	189	169	106	21	26	9	27	7	78.31	80.18	**79.23**
T-H	189	174	42	42	22	13	74	55	48,81	47,13	47,95
F-H	189	205	142	27	5	0	19	31	84.92	77.20	80.87
P-H	189	194	118	18	32	4	17	22	86.51	76.03	**80.93**

3.3.4.2 Experiments and Analysis

As mentioned before, the training set is used to adjust solution parameters. The slide images in the training set are screenshotted from the desktop stream of corresponding presentation or lecture recordings on tele-TASK.de. The resolution is 1024 × 768, but the image quality is actually quite limited, which causes a lot of OCR errors and further makes the proposed solution struggle. Detailed statistics can be found in "P-Tr" row of Table 3.1, where "Co." and "Part." are the short terms of "Correct" and "Partial" respectively.

The original documents in the benchmark set are saved in PDF format, on which ABBYY FineReader achieved 98% accuracy and won the competition [118]. However, in this evaluation FineReader needs to take the document images as input and apply its highly reputable commercial OCR tool [134, 135] on these images while detecting table. After transforming the benchmark set into high quality images, FineReader manages to achieve 89%, which is still very promising. This result, illustrated in Table 3.1, proves the reliability and effectiveness of FineReader table detection tool on traditional type of document. Tesseract works directly with images and its performance on benchmark set can also be referenced.

The slides in the test set are prepared in two formats: low-resolution screenshots (L) and high-resolution transformed images (H). The resolution of L-Images is still 1024 × 768, but the visual quality is generally better than those in the training set. When testing on L-Images, the proposed solution performs the best. H-Images are transformed directly from the digital files, either PPTX or PDF. There is no unified resolution, but all H-Images have higher resolution than 1024 × 768. Comparing with L-Images, All tested solutions perform better on H-Images, and the general accuracies of FineReader and proposed solution are almost the same.

From the statistics shown in Table 3.1, it is easy to figure out no matter with L-images or H-images, FineReader makes more detections than the proposed solution, but also includes more false positive detection. The proposed solution tends to make mistakes as over detection, while FineReader is more likely to make partial detections. In general, the proposed solution is proven to be better than FineReader in context of slide images, especially when the input image quality is limited. Tesseract is no match to either of these two.

By comparing the performances in benchmark set and test set, it is obvious that both Tesseract and FineReader perform less effective on slide images than traditional type of documents, which proves the importance of researching on slide-oriented table detection method. The general accuracy around 80% is not perfect, but enough for some fundamental applications like indexing table-inclusive slides or generating lecture outline.

3.4 Adaptive Tree-Structure Outline Generation

3.4.1 Framework

The basic framework of TOG (***T****ree-Structure* ***O****utline* ***G****eneration*) is depicted in Figure 3.5. It is a self-adaptive process based on multiple rounds, with an initial static round and several adaptive rounds. One round can be roughly divided into four major procedures: pre-processing, intra-slide layout analysis, inter-slides logic analysis and post-processing, each of which can be further subdivided into multiple steps. When there are external slide files available, in either PPTX or PDF format, an additional procedure of video-document synchronization can be

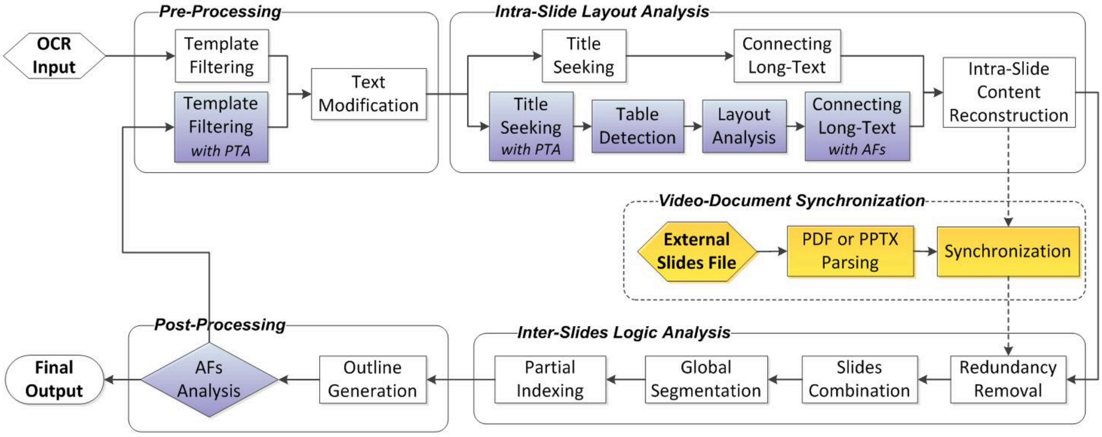


Figure 3.5: The framework of proposed TOG solution.

also activated. These procedures will be further illustrated in detail by following subsections accordingly.

In Figure 3.5, the initial static round can be considered as connecting all the steps with white background. ***A**daptive **F**eatures* (AFs) will then be analyzed at the end of the round. These features aim to describe some characteristics of the slide template used by the under-processing presentation or lecture, including:

- ◇ ***P**otential **T**itle **A**rea* (PTA): when more than 1/4 of the slides have their titles in same position, it can be left-aligned or centered.
- ◇ ***G**eneral **H**ierarchical **G**ap* (GHG): when more than 1/4 of the slides have same horizontal gaps between their level-1 and level-2 subtopics.
- ◇ ***L**ow **C**ase **S**tart* (LCS): when more than 30% of all outline items starting with lower-case letters.
- ◇ ***I**tem **B**ullet* (IB): when more than 20% of all outline items starting with a single-character word, which are probably misrecognized item bullets.

These features will affect several steps in adaptive rounds, along with a few new steps, which are marked with light blue background in Figure 3.5. At the end of each adaptive round, these features will be updated. If there are any changes, another round will be triggered. In order to avoid potential “dead loop”, the maximum of adaptive rounds is set to 3. Besides, the optional steps are marked with yellow background and dotted line.

3.4.2 Pre-Processing

3.4.2.1 Template Filtering

As already mentioned in Chapter 3.3, a slide can be simplified as a blank background and a bunch of text-lines after OCR. However, not all the text-lines are useful for the outline. Since most of the lecturers will edit their slides with affiliation-related templates, some text-lines are actually the affiliation logo, the lecturer's name or the foot line, which repeatedly appear in the same position, mainly at the edges of the slide, throughout the whole series. The information they carry is redundant and once misrecognized as a major part of the slide, it may drastically damage the content structure of the slide.

Thus, a searching scheme is applied to traverse the slide series and record those repeatedly appearing text-lines with same textual content and highly similar location parameters. If the accumulated appearance of such a "suspect" is beyond the threshold, which is decided by the total number of the slides, the bounding box of the suspected text-line will be marked as removal zone: any text-line locating in this zone will be removed, no matter whether its textual content is exactly the same as the original "suspect".

Sometimes a slide title, or at least a part of it, is also shared by several continuous slides. The above scheme may also put such titles under the risk of being removed. So in adaptive rounds, PTA is introduced to build a protection zone for potential titles. All text-lines locating inside or largely overlapping the PTA will directly bypass template filtering.

3.4.2.2 Text Modification

The general accuracy of the OCR program applied in TOG is around 85∼92%, which means, there are still approximately 10% of errors. It is impossible to predict how these errors may look like, but the worst case is an absolutely meaningless string consisting of random characters. No student would like to see such thing in lecture outlines, so it is necessary to detect and eliminate them.

The standards to define meaninglessness include extreme average word length (*either too long or too short*), extreme text-line sizes (*either too large or too small*), containing too many symbols, repeating same character for too many times, *etc.* A special dictionary is built for frequently used words with less than 3 characters, such as “is” and “a”, and common abbreviations like “IT”, to protect them from being deleted. Moreover, redundant spaces will also be removed.

3.4.3 Intra-Slide Layout Analysis

3.4.3.1 Title Seeking

Title is undoubtedly the most important component in a slide, especially in purpose of building a lecture outline. Generally, the title has three features: bold, upper-located and separated from other slide components. In TOG, up to 3 text-lines can be accepted as title candidates. They may occupy multiple rows and include potential subtitle (*extended title, not the transcript-like subtitle mentioned in Chapter 2 and Chapter 4*).

A title candidate must have an above-average height and vertically locates in the top 1/3 of the slide. The searching starts from the top, proceeds upside-down and adopts the principle of “first come first served”, but the text-lines locating too close to the slide edges will be ignored. When there is more than one candidate found, they should be adjacent either horizontally or vertically.

In adaptive rounds when PTA is available, the searching scheme will grant a priority to the text-lines locating in PTA. PTA has a strict limitation on vertical coordinates, but quite flexible horizontally, only indicating the position of the left end or the middle axis of the title area. If a text-line locates in this bar-shaped area, it will be acknowledged as fitting in. Once fitting in, no text-line outside the PTA will be further accepted. By this effort, some text components in the slide, especially the first line of the slide body, will be no longer mistakenly considered as the subtitle.

At last, all title candidates will be connected together from top to bottom and from left to right. The slide title is then saved and the text-lines involved will be hidden for all the later steps.

3.4.3.2 Table Detection

Table detection is only activated in adaptive rounds. Here the solution described in Chapter 3.3 is applied. The quality of the slide images in TOG is the same as the training set images in Chapter 3.3.4. Once a table is detected, all text-lines inside the table area will be removed and the table caption, if applicable, will remain for the later steps.

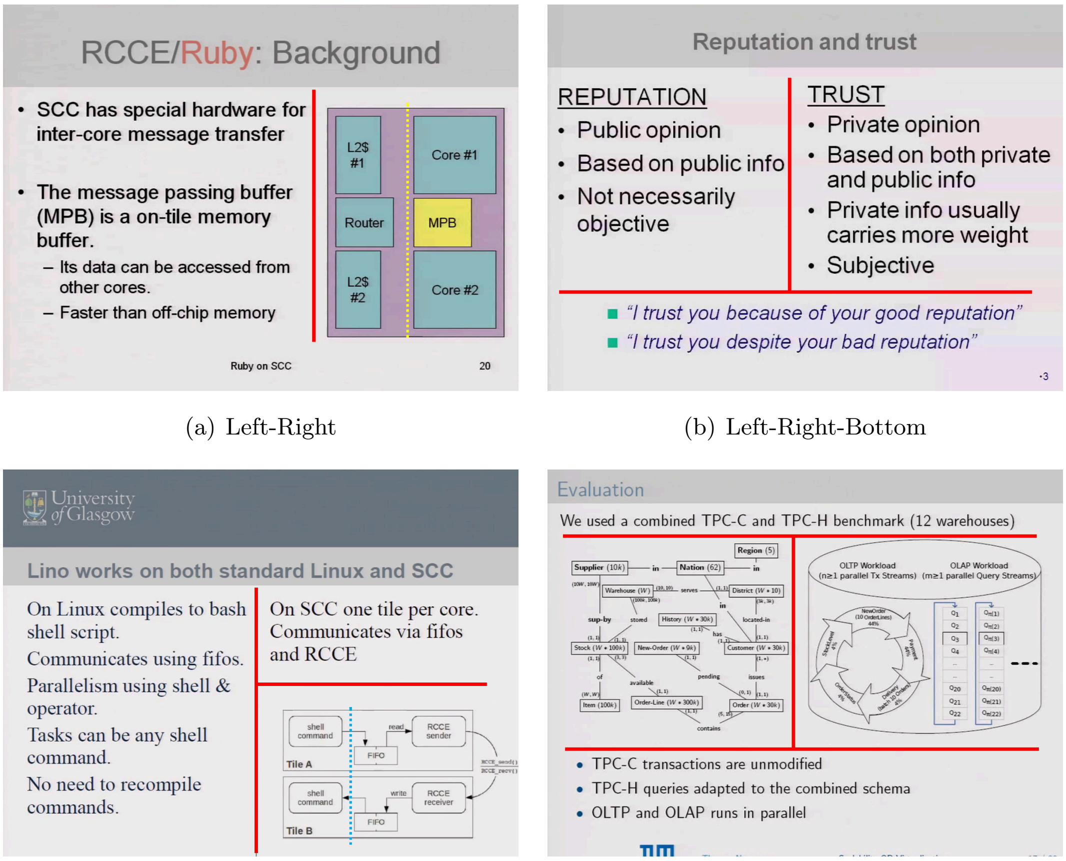
3.4.3.3 Page Layout Analysis

The layout of slides can be very diverse. In the initial static round, text-lines are loaded according to a fixed layout − the default single column "title and content" layout. Admittedly this default layout is most frequently used, but the incapability of the static version of TOG in dealing with diagrams, charts, two-column layout, *etc.*, results in inevitable quality decline in the outline generated. So in adaptive rounds, a method which can "smartly" analyze the layout per slide is developed to tackle the previously mentioned shortcomings.

After excluding the title and tables, it is not difficult to split text-lines into groups by setting up proper vertical and horizontal border lines, as the red solid lines shown in Figure 3.6 [1]. If a virtual rectangle is created to cover such a group, it becomes a block. A block can be either a text block which should be retained in the outline, or a diagram/chart block which needs to be neglected. The method of obtaining and distinguishing so-called blocks is presented below:

1. Attempt to find a middle line which horizontally divides all text-lines, except for the title, into left and right blocks. If there are more than one option, apply the one closest to the absolute middle.(*Figure 3.6-a and Figure 3.6-c*)
2. For every left-block or right-block, attempt to split it vertically into 2 sub blocks, as long as there is a huge line space inside the block. (*Right block in Figure 3.6-c*)
3. If step 1 failed, attempt top-left-right or left-right-bottom layout. In this case no further vertically splitting is applied. (*Figure 3.6-b*)
4. If step 3 still failed, attempt top-left-right-bottom layout. When this works, directly delete the left and right sub blocks found here. They are supposed to be diagrams. (*Figure 3.6-d*)

[1] Title areas of these slide images are grayed. The copyright of the example slides belongs to original authors: (a) Mr. Kouhei Ueno, (b) Prof. Audun Jøsang, (c) Mr. Paul Cockshott, (d) Prof. Thomas Neumann.



RCCE/Ruby: Background

- SCC has special hardware for inter-core message transfer
- The message passing buffer (MPB) is a on-tile memory buffer.
  - Its data can be accessed from other cores.
  - Faster than off-chip memory

Ruby on SCC 20

Reputation and trust

REPUTATION

- Public opinion
- Based on public info
- Not necessarily objective

TRUST

- Private opinion
- Based on both private and public info
- Private info usually carries more weight
- Subjective

- *"I trust you because of your good reputation"*
- *"I trust you despite your bad reputation"*

-3

University of Glasgow

Lino works on both standard Linux and SCC

On Linux compiles to bash shell script.
Communicates using fifos.
Parallelism using shell & operator.
Tasks can be any shell command.
No need to recompile commands.

On SCC one tile per core. Communicates via fifos and RCCE

Evaluation

We used a combined TPC-C and TPC-H benchmark (12 warehouses)

- TPC-C transactions are unmodified
- TPC-H queries adapted to the combined schema
- OLTP and OLAP runs in parallel

(a) Left-Right

(b) Left-Right-Bottom

(c) Left-Top-Bottom

(d) Top-Left-Right-Bottom

Figure 3.6: Some examples of different slide layouts and how they get dealt with.

5. Check all blocks or sub blocks whether they can be further split horizontally. If so, delete them as diagrams. (*Right block in Fig. 2-a and right-bottom sub block in Figure 3.6-c*)
6. Analyze all remaining blocks by their content. A block contains many digits, single words or not well-aligned will also be considered as a part of diagram or chart and gets deleted.

All remaining blocks are acknowledged as text blocks and will be treated as an independent text system in following steps. For those slides which cannot be split into blocks, all their text-lines are considered as a whole, in other words, an entire block.

3.4.3.4 Long-Text Connecting

When there is a long statement or description in the slide, it may occupy multiple rows, which happens frequently (*see Figure 3.2 and 3.6*). It is natural for human to read them continuously, but not for OCR program, by which several independent text-lines will be achieved in different rows. In purpose of making a user-friendly outline, items should contain full meaning to avoid misunderstanding, which means it is necessary to connect such long-text together.

Ideally there should be only one text-line in a row. If there are multiple text-lines locating in a same row, which can be determined by similar principle introduced in "row detection" of Chapter 3.3.1, a combination will be proposed if the horizontal gap between two text-lines is smaller than the width of 5 characters, which is calculated by the pixel-level total width and character-level total length of corresponding text-lines. Otherwise only the leftmost text-line in the row is retained, while others are removed. In static round, this method may cause information lost in a multi-column slide, but at least avoid mismatching. Table detection and page layout analysis would solve this problem in adaptive rounds.

Now from top to bottom, all text-lines in a block can be addressed as $t_1, t_2, ..., t_n$. In static round, if a combination is proposed for t_{i-1} and t_i, four requirements need to be satisfied:

- ◇ t_{i-1} starts with a upper-case letter or a number, while t_i starts with a lower-case letter.
- ◇ The line space between t_{i-1} and t_i is no larger than between t_i and t_{i+1}.
- ◇ Horizontally, the left-ends of t_{i-1} and t_i should be close, while the right end of t_i cannot exceed t_{i-1} too much.
- ◇ The width of t_{i-1} should not be too small, which is decided by the average of 4 adjacent text-lines.

Above requirements are designed based on the basic logic of word wrapping in text file. However, the slide templates diversify. Some templates do not require a capitalized sentence initial and some others have detectable bullets for slide items. So in adaptive round, with the help of adaptive features IB, LCS and

GHG, the long-text connecting approach could be updated. Instead of the hard-requirement scheme used in static round, a weight-threshold scheme is applied in adaptive rounds. The weights are derived from following decisive factors:

- ◇ When the adaptive feature IB is positive, a combination will be suggested if t_{i-1} has a bullet but t_i hasn't. As long as t_i has a bullet, the combination will be strongly opposed.
- ◇ The initial of a text-line can be upper-case, no-case (*e.g. digit*) or lower-case, with descending values. A combination will be suggested or opposed when the value of t_{i-1} is larger than t_i or not. And if the adaptive feature LCS is positive, the influential weight of this factor decreases.
- ◇ If the line space of t_{i-1} and t_i is way larger than their heights, or obviously larger than the line space of t_i and t_{i+1}, a combination is opposed. Otherwise it is slightly suggested.
- ◇ The left ends of t_{i-1} and t_i should be horizontally close if they belong to same sentence. Please note if the difference of their horizontal starting points fits the GHG, the combination will be vetoed.
- ◇ All text-lines sharing same horizontal starting point with t_{i-1} will be traversed and the widest one will be taken as reference. Only if the difference between the width of t_{i-1} and the reference is smaller than the width of the first word in t_i, the combination could be suggested.

All above factors will be quantified into weights, with "suggested" into positive values and "opposed" or "vetoed" into negative. Finally if the sum is above 0, a combination is applied. The parameters of the newly combined text-lines will then be adjusted to facilitate further combination attempts.

3.4.3.5 Intra-Slide Content Reconstruction

In this final step of intra-slide phase, a hierarchical content system will be reconstructed, which further serves as the tree-structure outline of certain slide. The main challenge here is to distinguish whether a text-line should be included in the content system and if yes, which level it belongs to. It will be done by analyzing the distribution and location logic of the text-lines.

In static round, the whole page is considered as an entire block and there is only one reconstruction method. Starting with searching in the left-top quarter of the slide, the goal is to find a large enough text-line whose left-end locates here. The definition of large enough is above the average text-line height and if there are multiple candidates, the leftmost one will be selected. Then the horizontal coordinate of its left-end is taken as the datum.

This datum might be either level-1 or level-2 location of the content system reconstructed, and the selected text-lines, along with all other text-lines aligning with the datum, will definitely be included in the system. Next step is to traverse all vertical adjacent text-lines of datum-aligned ones, in order to find potential indented horizontal coordinates of other levels, if they exist. Finally, up to 3 horizontal coordinates are confirmed and all text-lines aligning with any of them will be assigned with corresponding levels.

In adaptive rounds, the above method is taken as default, but there are more options. By taking text blocks as input, a second method specialized for center-aligned situation is applied on any text block containing less than 5 text-lines. In this case, the first text-line is directly set to level-1, and all others are set to level-2. If there are more than 5 text-lines in the block, a third method is introduced. It traverses all the text-lines to find out the most frequently used left-end coordinate as the datum, then seeking other levels as the default method.

For each block, both default method and corresponding alternative method are applied. By comparing their results, the method with more text-lines involved in the content system will be adopted. For those slides with multiple blocks, the content of different blocks will be combined together according to the order of top-left-right-bottom.

Additionally, an upgrading scheme is used in precaution of any logical disorder, such as a level-1 subtopic followed by a level-3 one. Now a slide can be represented by a purely textual slide unit, which consists of a title and an up-to-3-level intra-slide outline. In a slide unit, each text-line still carries the timing information inherited from OCR output. Although the location parameters of each text-line are no long needed, several special features, such as the title position and the horizontal coordinates to decide text-line level, will be stored per slide unit. Finally a slide unit will be marked as “well-organized” or “ill-organized”, based on how many text-lines are not included in the content system reconstructed. This attribute is useful in following inter-slides phase.

3.4.4 Video-Document Synchronization

3.4.4.1 File Parsing

When the external slide file is available, it is a great chance to improve the textual accuracy of the outline generated. Generally, there are two different formats of the external slide file: PDF and PPTX. Different approaches are developed for them respectively, both of which are parsed based on Apache toolkits[1].

In context of TOG, the parsing result of PDF file can be actually considered as error-free character-level OCR result. The task here is to cluster characters into words and then cluster words into text-lines. Vertical line space is naturally the sign of text-line change, and for the adjacent characters in same row, the crucial factor is the horizontal gap. The maximum width of adjacent two characters is taken as datum. If the gap is larger than 1/3 of the datum, a space will be inserted as the word boundary. If the gap is larger than 1.5 times of the datum, it is acknowledged as text-line border. These thresholds work perfectly in TOG. When clustering finishes, these PDF originated text-lines will be fed into preprocessing and intra-slide layout analysis to create a parallel sequence of textual slide units.

When the file format is PPTX, the approach is much easier. After parsing by Apache toolkit, the slide components are already categorized into title, text paragraph, table, image, *etc.* In context of TOG, only title and text paragraphs need to be retained. Furthermore, the text paragraphs are also hierarchically saved. So by simply extracting necessary information from the parsing result, a complete sequence of textual slide units can be directly achieved.

3.4.4.2 Synchronization

When there are two sequences of slide units available for one presentation, it is natural to consider generalizing the advantages from both of them. The slide units from external file have better textual accuracy, but only the slide units

[1] https://poi.apache.org/slideshow/xslf-cookbook.html & https://pdfbox.apache.org/

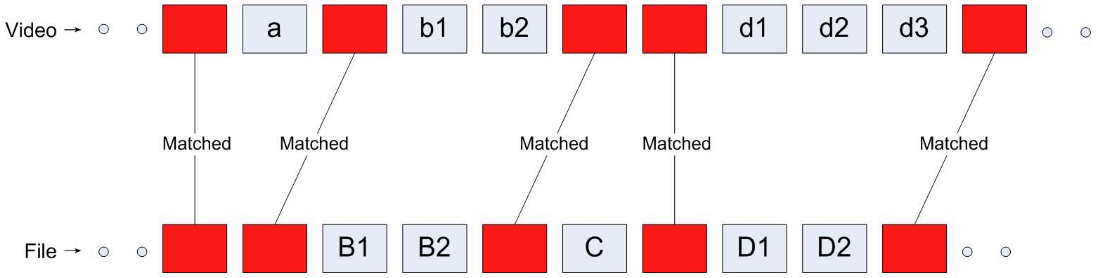


Figure 3.7: An example of Video-File Synchronization

originated from the desktop stream video contain the timing information which can synchronize the slide pages with video displaying. Since external file is only optional and not always available, the video-based sequence will be taken as baseline, textually updated by the file-based sequence, and further passed to following procedures. This arrangement would preserve the simplicity and compatibility of inter-slides phase.

Generally, the total number of slide units in video-based sequence is different from the file-based sequence. In order to update the text, mapping slides within two sequences is necessary. A mapping matrix would be created by calculating the textual similarity between slide pairs, with four possible statuses: content-same, title-same, title-similar and unrelated. The definition of "content-same" and "title-similar" will be introduced later in Chapter 3.4.5, while the others can be understood literally.

Based on the mapping matrix, two sequences will be first segmented in parallel by "Matched" content-same slide pairs (*red pairs in Figure 3.7*). If occasionally a slide unit in video-based sequence is content-same with multiple slide units in file-based sequence, the pair which can make the lengths of corresponding segments mostly close will be applied. After updating the content in these content-same slide units, it proceeds into segment-level.

If the lengths of corresponding segments are the same (*B/b sequences in Figure 3.7*), the content updating goes one by one. Else if the length of file-based segment is 0, redundant slide units in video-based sequence will be removed (*slide 'a' in Figure 3.7*). Else if the length of video-based segment is 0, extra slide units in file-based sequence is ignored (*slide 'C' in Figure 3.7*). Otherwise, as the D/d sequences, the segments will be further sub-segmented by title-same slide pairs, with exactly same follow-up processing. If needed, title-similar slide pairs can be used in a third round, on those parallel sub-segments with unequal lengths. After all possible content updating, the file-based sequence will be deleted.

3.4.5 Inter-Slides Logic Analysis

3.4.5.1 Redundancy Removal

Before analyzing the logical correlation between different slides, a step to remove redundancy is necessary. Redundancy is actually what every lecturer wants to avoid, but in real-time presentation, it is quite common to roll back to a previous slide when some further explanation is needed, or switch off the slides for a demo, or just simply misoperate. All these behaviors will be recorded by the desktop stream of the lecture video and reflected in the video-based slide unit sequence as redundancy.

The prerequisite to remove such redundancy is detecting repeated slides. All texts within a slide unit, including title, intra-slide outline and unattached text-lines, will first be connected into a single long string. Then both ***L**evenshtein* ***D**istance* ***R**ate* (LDR) and ***S**hared* ***W**ord* ***R**ate* (SWR) are calculated between each pair of such strings. Please note LDR is calculated against the longer string, while SWR is against the shorter string. The decision scheme is explained below:

1. If LDR $<$ 30% and SWR $>$ 70%, marked as repeated.
2. Else if LDR $<$ 50%, SWR $>$ 60%, the titles are the similar (LDR-T $<$ 20%) and the string is longer than 100 characters, marked as repeated.
3. Else if LDR $<$ 60%, SWR $>$ 70%, the titles are exactly the same and the string is longer than 200 characters, marked as repeated.

After traversing all possible pairs of slides, the repeated ones will be saved in groups, since it is very likely for a repeated slide to appear more than twice. Above scheme is also applied as the definition of "content-same" and "title-similar" in Chapter 3.4.4.2.

Based on the repeated slides detected, the redundancy caused by the lecturer's behavior of "rolling" slides can be dealt with. Suppose there is an original slide sequence "A-B-C-D", when the lecturer rolls back from slide C to A and then

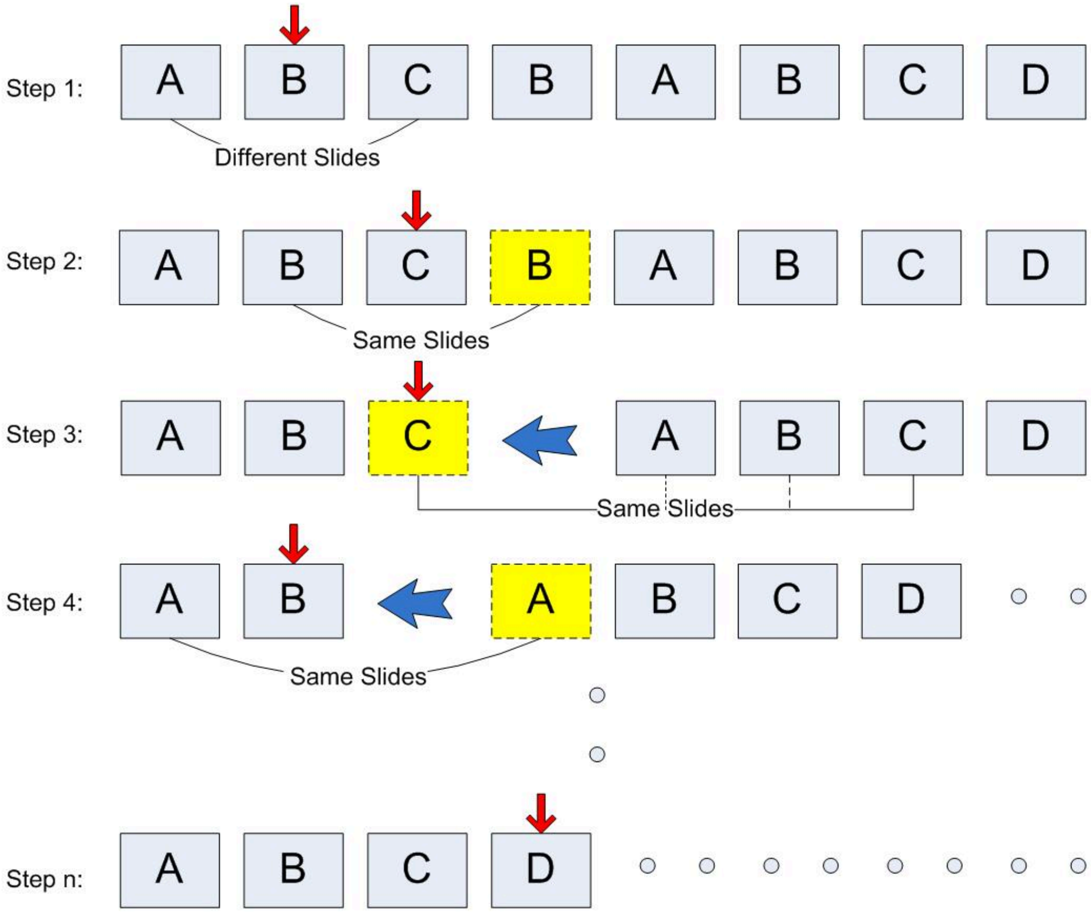


Figure 3.8: An example of the process of 'rolling' slides removal

move forward to D, the actually achieved sequence should be "A-B-C-B-A-B-C-D", and A, B, C can be found in the repeated slides groups. Figure 3.8 illustrates how the sequence could be simplified.

The process starts from the second slide in the sequence, by checking whether the previous slide and next slide are repeated. If not (*Step 1 in Figure 3.8*), the current slide pointer simply moves forward to the next slide and check again. If they are repeated, the "next" slide is removed while the previous slide is retained (*Step 2*). Then the current slide will also be checked on whether involved in any repeated groups (*Step 3, with 'C' as current slide*). If so, this current slide will also be removed and the pointer move backward to the previous slide (*Step 4*), otherwise it moves forward. By repeating above steps, the expected basic sequence “A-B-C-D” could be achieved.

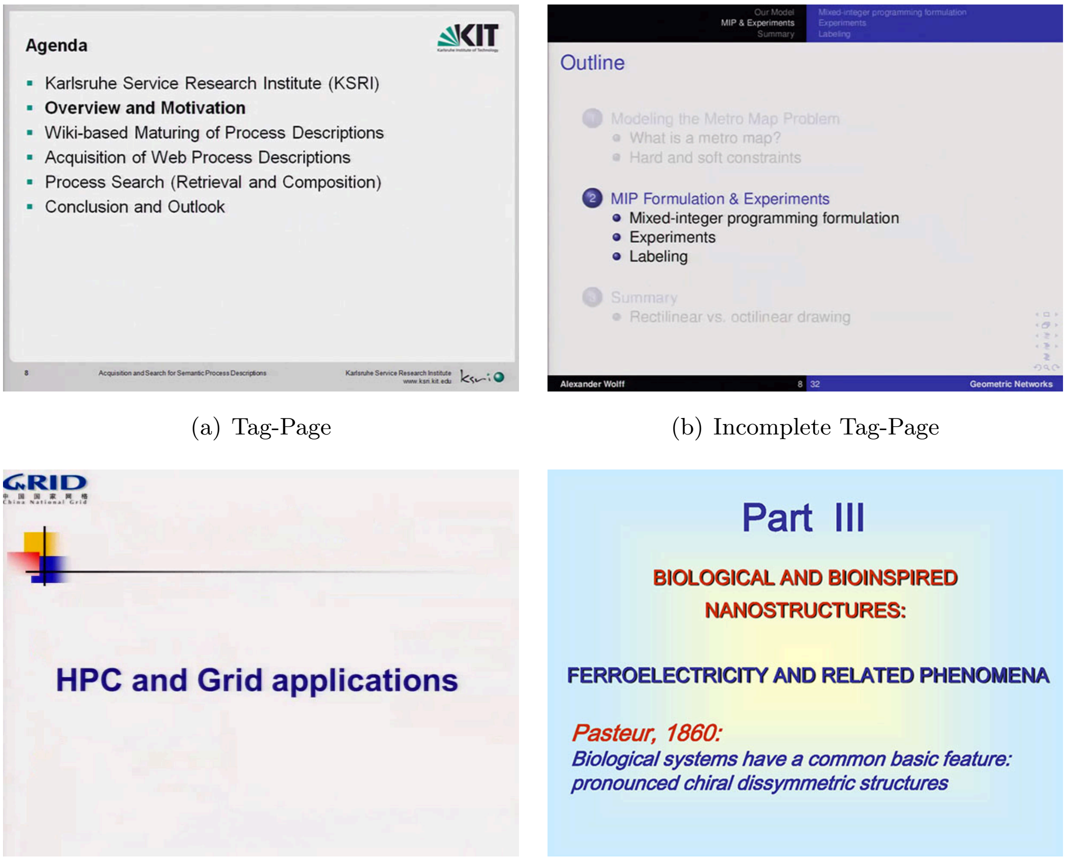
The next type of redundancy to be removed is called “live-show”, which is caused by the lecturer switching off the slides to play a demo video or show some products. The live-show could improve the vividness of the lecture but contribute nothing to the outline. A clue to identify live-show is that the same slide appears both before switching off and after switching back, which will be detected as repeated slides. And the slide units in between, which are actually a bunch of key frames extracted from approximate natural video, are highly probable to be “flashing”, which means the duration is very short, and ill-organized, because OCR can hardly capture anything meaningful from these non-slide frames.

According to these facts, each interval between adjacent repeated slides will be tested. If over half of slides in between are ill-organized and the average slide duration is less than 5 seconds, or over half of them are flashing by less than 1 second, this interval will be marked as live-show and removed entirely, as well as the repeated slide at the interval end.

3.4.5.2 Slides Combination

In lecture slides, one important topic may cover several continuous slides. These slides share a same title, sometimes with additional serial number. In purpose of generating lecture outline, combining them together will improve the quality, since repeated titles are also somehow redundant. Therefore, if two continuous slides have exactly same title, or the only difference between their titles are the serial numbers, such as “(1/3)”, “<b>”, “III”, *etc.*, they will be combined.

When combining, the intra-slide outline of the second slide will be added at the end of the first one. But if the content of the first slide can be fully covered by the second slide, which might cause by progressive displayed slide being detected as several independent slides, the overlapped part will be removed. In principle, the hierarchies of corresponding text-lines will be unchanged, including the unattached ones.



Agenda

- Karlsruhe Service Research Institute (KSRI)
- **Overview and Motivation**
- Wiki-based Maturing of Process Descriptions
- Acquisition of Web Process Descriptions
- Process Search (Retrieval and Composition)
- Conclusion and Outlook

8 Acquisition and Search for Semantic Process Descriptions Karlsruhe Service Research Institute www.ksri.kit.edu

(a) Tag-Page

Our Model
MIP & Experiments
Summary

Mixed-integer programming formulation
Experiments
Labeling

Outline

1. Modeling the Metro Map Problem
   - What is a metro map?
   - Hard and soft constraints
2. MIP Formulation & Experiments
   - Mixed-integer programming formulation
   - Experiments
   - Labeling
3. Summary
   - Rectilinear vs. octilinear drawing

Alexander Wolff 8 32 Geometric Networks

(b) Incomplete Tag-Page

HPC and Grid applications

(c) Split-Page

Part III

BIOLOGICAL AND BIOINSPIRED NANOSTRUCTURES:

FERROELECTRICITY AND RELATED PHENOMENA

Pasteur, 1860:
Biological systems have a common basic feature: pronounced chiral dissymmetric structures

(d) Section-Page

Figure 3.9: Some examples of 'border' slides

3.4.5.3 Global Segmentation

In many cases the content of a lecture is not lineally arranged. Alternatively, quite a lot of lectures consists of several segments, each of which focuses on a subtopic. In the manifestation of slides, the creator sometimes makes these segments very obvious, by inserting several special slides between the subtopics, which are addressed as "border" slides. When the border slides do exist, they provide a clear clue that how the lecture is constructed, so what needs to be done is just correctly locating them. Figure 3.9[1] shows a few examples of so-called "border" slides with different types.

[1] The copyright of the example slides belongs to original authors: (a) Prof. Rudi Studer, (b) Prof. Alexander Wolff, (c) Prof. Depei Qian, (d) Prof. Gil Rosenman.

Slide like Figure 3.9-a is addressed as ***tag-page***, which is roughly a simplest outline of the whole lecture. When existing, it definitely appears multiple times, has a highly recognizable title, such as “Agenda”, “Outline”, “Topics”, *etc.*, and its content is in fact a list of all subtopics. In each appearance, one of its text-lines will be highlighted and indicate which subtopic is coming. By locating these tag-pages in the repeated slides groups, the boundaries between subtopics can be precisely confirmed.

However, OCR cannot distinguish the highlighted text-line from the others, and there might be more text-lines in a tag-page than the total number of subtopics in the lecture (*e.g. sub-subtopics, just as Figure 3.9-b*). Thus, a synchronization scheme similar to what has been introduced in Chapter 3.4.4.2 is applied. But this time the baseline sequence is the text-lines in tag-page, and the reference sequence is the list of slide titles between tag-pages. Although it rarely happens, if there are more subtopics than text-lines in tag-pages, the extra subtopic will be named as “Subtopic No. X”, where X is decided by the position of this subtopic.

There is a variant of tag-page, the ***incomplete tag-page***, as Figure 3.9-b. It serves exactly the same as a tag-page, but instead of highlighting current subtopic, it visually “hides” the others. In this case, the hidden subtopics are undetectable by OCR, so the incomplete tag-pages cannot be found in repeated slides groups. Luckily, their highly recognizable title still appears multiple times and can be located. After restoring the subtopic list by combining their content, the incomplete tag-pages will be processed just as tag-pages.

Another type of widely used border slide is the ***split-page***, Figure 3.9-c gives an example. The only content in a split-page is the subtopic of following slides and generally, it locates in the center of the slide rather than the title position. It means a split-page after the previous intra-slide procedure would have no title and only less than three text-lines. If multiple such slide units can be found in the sequence and they are sparsely distributed, they will be acknowledged as split-pages and used as the boundaries of global subtopics.

Section-page is also adopted sometimes. Just like what Figure 3.9-d shows, a section-page is more than a border and perhaps has all features a common slide may have: expressing definitions, explaining algorithms or showing charts and images. The only difference of a section-page is in the title, which contains some special words indicating its identity as a border slide, such as “Topic 1”, “Theme B”, “PART III”. If more than two such section-pages with same format are found in the slide sequence, they would be taken as subtopic boundaries.

These four types of border slides have a priority order just as they are introduced, which functions when there are occasionally more than one type is detected. If the last slide of the lecture has special title like “Thanks”, “Contact” or “Summary”, it will not be included in the last subtopic. By now, all remaining repeated slides will finally be deleted.

3.4.5.4 Partial Indexing

For those lectures without recognizable border slides, or when the globally segmented subtopics are still too general, partial indexing is implemented. Unlike global segmentation, which disassembles subtopics on whole lecture scale and goes top-down, partial indexing attempts to explore potential logical relation within several adjacent slides and build up subtopics in a bottom-up way. Generally, partial indexing affects in two forms: index-page and virtual index-page.

An ***index-page*** is somehow a regional tag-page. The content of an index-page is a preview of several following slides, and in extreme case, an index-page could be a collection of several following titles. So it is natural to consider them as closely related and belonging to a same subtopic. Since index-page has no special title, the searching scheme will be applied throughout every global segment and the slide sequence before, or the whole lecture when there is no border slide available. If half of all text-lines in a slide can match following titles, it will be acknowledged as an index-page and then the text-lines will be synchronized with following slides just like what is done with tag-page.

Virtual index-page derives from a series of continuous slides which share same keyword or prefix in their titles. In this case, these slides are supposed to describe different aspects of a certain theme, which is exactly the shared keyword or prefix. In order to avoid too many false positive cases, only nouns are taken in counting, and the frequency of the keyword must be higher than 60% in the counting interval. Once confirmed, a new slide unit will be created with the shared keyword or prefix as title and the following slide titles in the interval as text-lines,

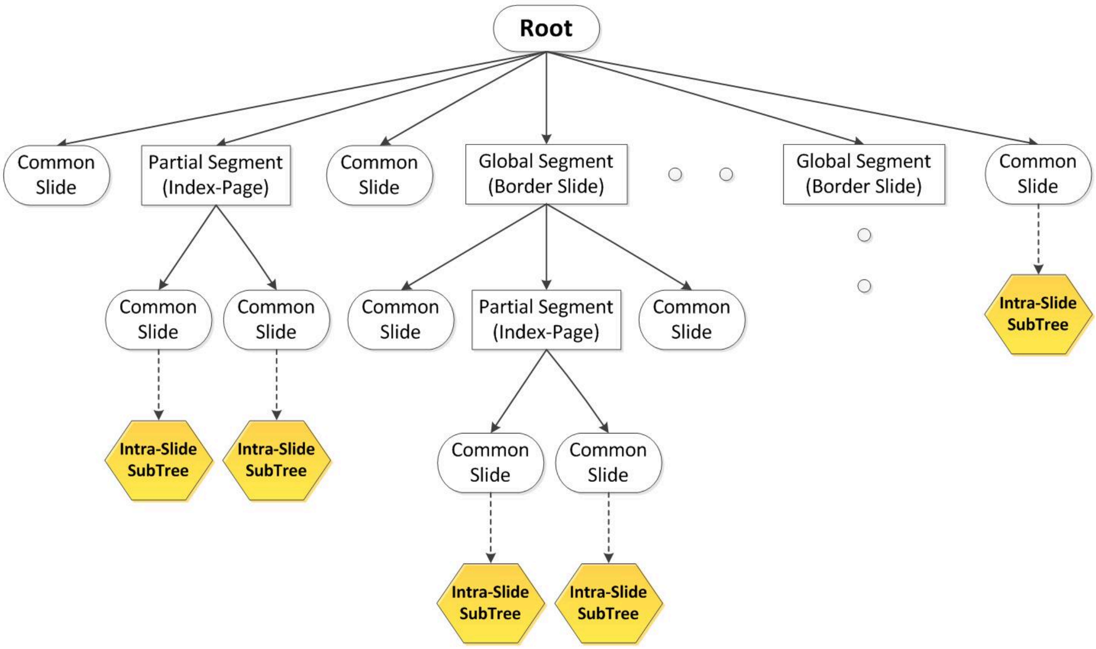


Figure 3.10: A sketch map of the tree-structure lecture outline generated.

just as an index-page. This newly created virtual index-page will be then inserted at the beginning of the interval. The searching of virtual index-pages only applies on slides which are not included in any global or partial segments previously.

Now based on these logical relations explored, the whole sequence of slide can be arranged in tree-structure, also up to 3 levels. Figure 3.10 is a sketch map of such slide trees (*white background components and solid lines only*). The root can be considered as the lecture title. Then common slides, border slides and index-pages directly under the root are in level-1. Level-2 slides include common slides and index-pages under a global or partial segment. Only common slides under a level-2 index-page can be placed in level-3.

3.4.6 Post-Processing

3.4.6.1 Outline Generation

At the end of each round, the complete lecture outline can be created by considering both intra-slide and inter-slides analysis result. All remaining text-lines which are not included in any intra-slide content system will be removed. The slide title will be set to level-0 and other text-lines keep their hierarchies ranged 1∼3. The slide hierarchy has also been decided already, with the same range 1∼3. Then the slide title will act as the root of the related intra-slide content subtree and take the position of corresponding "common slide" in Figure 3.10, the leaf node of the slide tree.

Besides, the positions of global or partial segments will be taken by the corresponding text-lines in border slides or index-pages which indicate subtopics. As a result, a pure textual tree-structure lecture outline is achieved. The final hierarchy of each outline item is calculated by a simple addition, ranged 1∼6. Both intra-slide hierarchy and final hierarchy are saved by each outline item, in order to facilitate different configurations when presented to learners. Timing information is still saved in the outline items to facilitate any further applications related to navigation.

3.4.6.2 Adaptive Feature Analysis

Another task in post-processing is to analyze or/and update the adaptive features. The title position of each slide is recorded and the repeatedly used positions are saved as PTAs. The gap between level-1 and level-2 text-lines is also recorded and GHG is used to save the value if there is a traceable pattern in gaps. Both these two thresholds are set to 1/4.

The Boolean features LCS and IB derive from the statistics of the final outline. When more than 30% of the outline items begin with lower-case letters, LCS will be set to "true". If an outline item starts illogically with a single character, such as "o", it will be considered as a misrecoginized bullet point. When over 20% of outline items initiate with such bullet points, IB will be activated.

Except for the initial round, all four updated features will be compared with "themselves" from previous round. If there is no change, the outline achieved in this round will be taken as the final output. Otherwise a new adaptive round will be launched with these updated features. But as already mentioned in Chapter 3.4.1, the upper limit of adaptive rounds is 3, in order to avoid infinite loop.

Table 3.2: Intra-slide phase accuracy report.

	ID	Character Aspect			Item Aspect					
		Length	L.D.	Prec.	G.T.	Hit	Rec.	All	Correct	Prec.
Static Round	5626	2305	393	83.0%	66	50	75.8%	82	36	43.9%
	5759	5596	1625	71.0%	208	162	77.9%	215	128	59.5%
	6011	6888	596	91.3%	123	115	93.5%	132	101.5	76.9%
	6031	6103	961	84.3%	158	129	81.6%	139	114	82.0%
	6102	5637	1223	78.3%	162	147	90.7%	184	137	74.5%
	6106	4417	2800	36.6%	107	87	81.3%	169	68.5	40.5%
	6196	7742	2585	66.6%	268	152	56.7%	194	132.5	68.3%
	6201	3381	786	76.8%	118	104	88.1%	133	100.5	75.6%
	6261	4268	1273	70.2%	132	98	74.2%	132	93.5	70.8%
	6266	2569	272	89.4%	65	63	96.9%	81	61	75.3%
	6663	4014	245	93.9%	98	94	95.9%	113	89.5	79.2%
	7314	2820	1352	52.1%	83	63	75.9%	109	60	55.0%
	All	55740	14111	**74.7%**	1588	1264	**79.6%**	1683	1122	**66.7%**
Adaptive Solution	5626	2305	205	91.1%	66	64	97.0%	68	58.5	86.0%
	5759	5596	755	86.5%	208	185	88.9%	205	156	76.1%
	6011	6888	612	91.1%	123	110	89.4%	124	98	79.0%
	6031	6103	515	91.6%	158	154	97.5%	166	146.5	88.3%
	6102	5637	1263	77.6%	162	153	94.4%	184	141	76.6%
	6106	4417	1074	75.7%	107	102	95.3%	139	89	64.0%
	6196	7742	1767	77.2%	268	230	85.8%	257	220	85.6%
	6201	3381	328	90.3%	118	112	94.9%	117	111	94.9%
	6261	4268	506	88.1%	132	115	87.1%	136	105.5	76.1%
	6266	2569	302	88.2%	65	63	96.9%	76	60.5	79.6%
	6663	4014	205	94.9%	98	91	92.9%	94	87	92.6%
	7314	2820	407	66.6%	83	68	81.9%	78	59.5	76.3%
	All	55740	7939	**85.8%**	1588	1447	**91.1%**	1644	1330.5	**80.9%**

3.4.7 Evaluation

In order to evaluate the quality of lecture outline generated, a test set with 12 complete e-lectures is prepared. These lectures were given by 12 different lecturers. A total number of 354 pages of original slides are supposed to be

Table 3.3: Inter-slides phase accuracy report.

ID	T.S.	G.T.	Static		Adaptive	
			Correct	Accuracy	Correct	Accuracy
5626	17	13	10	76.9%	10	76.9%
5759	45	20	5.5	27.5%	19.5	97.5%
6011	18	10	5	50.0%	8	80.0%
6031	22	20	7	35.0%	19	95.0%
6102	36	30	26.5	88.3%	28	93.3%
6106	28	28	25	89.3%	25	89.3%
6196	81	31	19	61.3%	21.5	69.4%
6201	25	25	22	88.0%	23	92.0%
6261	27	20	12	60.0%	14	70.0%
6266	18	18	12.5	69.4%	14	77.8%
6663	20	20	19	95.0%	19	95.0%
7314	17	18	12	66.7%	10.5	58.3%
All	354	253	175.5	**69.4%**	211.5	**83.6%**

extracted from the desktop stream of 437 minutes of lecture videos, by which the diversity of the dataset could be assured. A lecture ID is used to identify certain lectures, and all these lectures are publicly available on tele-TASK platform[1]. The ground-truth is manually created. Since there are no external slides files available for most of these lectures, the performances reported are totally based on the screenshotted slide images. Both the outputs of initial static round and final round are reported.

The performance of intra-slide phase has two aspects: characters and items. In character aspect, all remaining textual data within a slide are connected together as a long string, and then a Levenshtein distance (L.D.) is calculated between this string and the ground-truth (G.T.), which is similar to the process of detecting repeated slides in Chapter 3.4.5.1. Theoretically the smaller the L.D. is, the higher the precision reaches. Then in item aspect, including the title and all in-system text-lines, both readability and hierarchy are tested and value 0.5 respectively. By comparing with the G.T., both recall and precision can be measured. Please note minor differences in characters leading no misunderstanding

[1] The lecture (ID = ‘id’) is in http://www.tele-task.de/archive/lecture/overview/id/

Table 3.4: General Accuracies

	Intra-Slide (A_1)			Inter-Slides (A_2)	A_{final}
	Character (P_C)	Item			
		R_I	P_I		
Static	74.7%	79.6%	66.7%	69.4%	**71.5%**
Adaptive	85.8%	91.1%	80.9%	83.6%	**84.7%**

will be ignored in item aspect, since they have already affected the precision in character aspect. Statistics can be found in Table 3.2.

The second phase of evaluation focuses on the inter-slides logic. Slide title will represent the whole slide, with a string and a hierarchy ranged 1∼3. Similar to the item-aspect intra-slide evaluation, the string and the hierarchy weight 50% each. Table 3.3 shows the result. Total slides (T.S.) indicates how many slides have been extracted from the video originally, which in many cases differs from the G.T., due to the expected changes in procedure of inter-slides logic analysis.

A general accuracy will be calculated by covering all aspects, as presented in equation (3.2). The general intra-slide item-level accuracy is achieved by applying F-measure (*harmonic mean*) on recall (R_I) and precision (P_I). Then the G-measure (*geometric mean*) of both item-level accuracy and character-level precision (P_C) is taken as the general intra-slide accuracy. Finally the G-measure of both intra-slide and inter-slides accuracies (A_1 *and* A_2) is adopted as the general accuracy of proposed TOG.

$$A_{final} = \sqrt{A_2 \cdot \sqrt{P_C \cdot \frac{R_I \cdot P_I}{R_I + P_I}}} \tag{3.2}$$

From Table 3.4 it is easy to figure out that the general accuracy of the outline generated can reach approximately 85%, which is fairly good. If offered to the e-learning portal users, this outline with 85% accuracy contains enough correct information and could provide a trustworthy overview of the whole lecture. Considering the input OCR accuracy is also around 85%∼92%, the error rate cause by deficiency in TOG is actually even lower.

3.5 Lecture Video Segmentation

3.5.1 Method

As already mentioned in Chapter 3.2, it is natural to take the slide transition events to segment those lecture videos with synchronized slides. These single-slide segments would be great in purpose of navigation, just as shown in Figure 1.1, but on the other hand, perhaps they are too fragmented to represent comparatively integrated subtopics. In previous TOG process, logical relations between slides have been analyzed by continuous slide combination, global segmentation and partial indexing. The analysis result is reflected in the final outline generated. Since the timing information is also reserved in the outline, the video segmentation task can be applied by reverse outline parsing.

Here the intra-slide content is not important, so a title-only outline will be extracted and further parsed. Apparently, a slide in level-2 or level-3 belongs to a recognized subtopic leading by the closest previous slide of level-1. Based on this principle, involved slides can be grouped together and naturally taken as a segment. The logical basis of such segments is firm and strong, and the title of such segments will be the unique level-1 outline item inside. These segments are addressed as ***L**ogical **S**egment* (LS) and further divided into ***G**lobally **L**ogical **S**egment* (GLS) and ***P**artially **L**ogical **S**egment* (PLS), according to their origins respectively.

When some slides are not included in any GLS or PLS, a default segmentation method based on duration will be applied. First, expected segment duration is calculated by the minimum of 1/4 lecture length and average LS length, if applicable. Then continuous free slides sequences are traversed as well as the total duration accumulates. Once the sum surpass the expected segment duration, a new ***T**ime **S**egment* (TS) will be created. After that, the traversal proceeds with the sum cleared until reaching the boundary of a LS or the end of the lecture.

If there is only one slide in the queue when the traversal reaches the end, this slide will be combined in the previous segment, rather than creating a new n-TS with only one slide. The title of the slide which has the longest duration in corresponding TS will be also taken as the segment title. If a TS is adjacent to at least one LS, it will be identified as "positive TS (p-TS)", because more or less

Table 3.5: Basic statistics about outline-based lecture video segmentation.

ID	Border	Total	GLS	PLS	p-TS	n-TS	Ave Duration
5373	Yes	8	7	0	1	0	6:50
5626	No	5	0	1	0	4	8:05
5724	No	6	0	1	1	4	5:03
6011	No	3	0	1	2	0	10:59
6021	No	4	0	2	2	0	8:27
6027	No	5	0	2	3	0	5:51
6031	No	5	0	2	3	0	6:06
6098	Yes	6	4	0	2	0	7:58
6102	Yes	6	3	1	2	0	3:19
6104	No	5	0	1	1	3	7:12
6106	Yes	5	4	0	1	0	6:22
6196	No	8	0	4	4	0	7:20
6201	No	4	0	2	2	0	13:06
6212	Yes	5	2	1	2	0	12:26
6225	Yes	4	2	0	2	0	12:08
6261	No	6	0	2	4	0	8:02
6266	No	3	0	1	2	0	6:38
6415	Yes	7	5	0	2	0	6:48
6663	Yes	6	4	0	2	0	4:03
7167	Yes	5	3	1	1	0	5:24

it has some logical basis. All the others will be named as “negative TS (n-TS)”. Obviously the less n-TS there is, the better segmentation works.

3.5.2 Evaluation

The test dataset consists of 20 lecture videos, which partially coincides the 12 lectures used in outline evaluation. Similarly, these videos can be found on tele-TASK platform by their lecture ID. Table 3.5 depicts the basic statistics of the segmentation result. “Border” means whether the lecture contains border slides which are defined in Chapter 3.4.5.3. Then the numbers of totally achieved segments, GLS, PLS, p-TS and n-TS is listed, followed by the average segment duration.

Table 3.6: Proportion analysis of lecture segments.

Slide Type	GLS	PLS	p-TS	n-TS	Ave Duration
with Border	65.4%	5.8%	28.8%	0%	7:15
w/o Border	–	35.8%	39.6%	24.5%	7:53
All	32.1%	20.8%	34.9%	12.3%	7:36

According to Table 3.5, the lecture videos are generally split into 3∼8 segments with the average duration controlled in 3∼13 minutes. Logical relations between slides more or less can be found in all testing lectures. Only 3 of 20 lectures achieve n-TS which is unwanted.

More specifically, in all 9 lectures which actually contain border slides, they are successfully detected. According to the analysis shown in Table 3.6, over 65% of segments in those lectures are GLS, in addition with a few PLS, which generally comes from the virtual index-pages which locate before the first border slide, makes approximate 70% of segments having strong logical basis. In other lectures, 35% of segments also belong to PLS. Together with the related p-TS, the ratio of n-TS is limited to less than 1/4.

In general, almost 90% of the segments are achieved by some logical reasons, including about 1/3 of GLS, 1/5 of PLS and 1/3 of p-TS, while the average duration is 7:36. As a follow-up application of TOG, the proposed lecture video segmentation approach achieves fairly promising result with very limited resources.

3.6 Chapter Summary

In this chapter, an adaptive process of generating tree-structure lecture outline (TOG) is presented, along with a sub-application of table detection and a follow-up application of lecture video segmentation. All these works are based on the desktop stream video of lecture recordings, which generally captures the lecture slides, with STD and OCR as pretreatment.

The proposed table detection method deals with the diverse table formats in slide images by ignoring all shortcuts and simply starting with detecting rows and columns. After locating, confirming and expanding the table area which is generally built on the intersected text-lines, the proposed solution slightly outperforms the state-of-the-art commercial software ABBYY FineReader and significantly surpasses the open-source tool Tesseract in task of detecting tables from slide images.

Consisting of pre-processing, intra-slide layout analysis, inter-slides logic analysis and post-processing, in addition with optional video-document synchronization, the proposed TOG is able to create up-to-6-level lecture outline with the general accuracy of 85%. Once offered to the online learners, this accuracy would basically satisfy the requirement of the supposed functions a lecture outline should have, including preview, navigation and retrieval.

As a follow-up application, lecture video segmentation can be implemented by reversely parsing the lecture outline already generated. The evaluation result shows that roughly 90% of the segments achieved are more or less supported by logical basis and the average segment duration, 7:36, is in general appropriate.

By proposing above comprehensive solution, the information contained in the lecture slides, or in other words, the desktop stream video, is highly utilized. The output of the proposed solution can be easily implemented in e-learning portal and facilitate its users.

Chapter 4

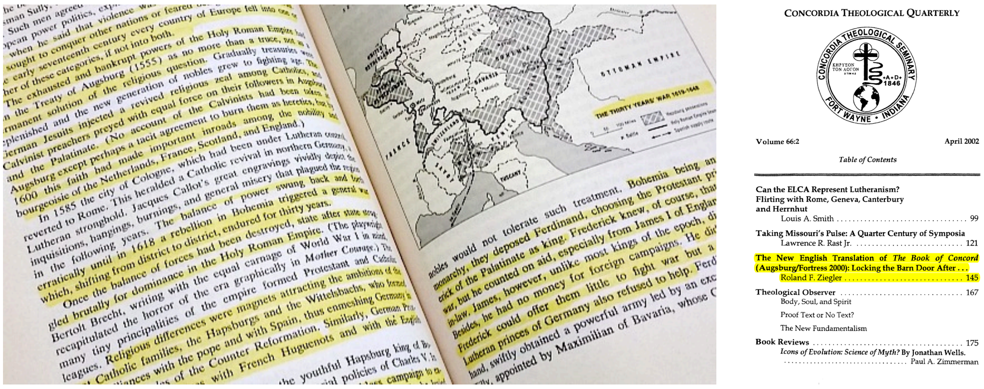
Lecture Highlighting

4.1 Motivation

Many people like using a marker to highlight books while reading, especially students with textbooks in hand [136]. Research shows that properly highlighted contents indeed support understanding [137]. It might be the reason why quite a lot of book authors already highlight the key concepts, features or equations in their books, and more are requested to do so [138, 139]. Generally, there are two types of highlighting: content highlighting and table of contents highlighting (*see Figure 4.1*). The former mostly emphasizes on certain sentences, while the latter works in a larger scale, indicating which chapter or subsection should be given special attention.

Not only popular with traditional paper books, the highlighting function is also welcomed in the era of e-books [140, 141]. It is widely implemented in various e-book applications and taken as the sign of "active reading" [142, 143]. In this case, although a marker is no longer required, highlighting is still based on book-like textual materials. However, what if there is nothing textual, such as attending a lecture without textbook, does that make sense to highlight the lecture?

The answer is yes. In a lecture, there are always some key-points among general introduction. These key-points might be expressed in certain definitions, illustrations, functions, applications, *etc.*, which might be more important to students than other contents in the lecture. Fortunately, good teachers are aware



CONCORDIA THEOLOGICAL QUARTERLY

Volume 66:2 April 2002

Table of Contents

Can the ELCA Represent Lutheranism? Flirting with Rome, Geneva, Canterbury and Herrnhut
Louis A. Smith 99

Taking Missouri's Pulse: A Quarter Century of Symposia
Lawrence R. Rast Jr. 121

The New English Translation of *The Book of Concord* (Augsburg/Fortress 2000): Locking the Barn Door After . . .
Roland F. Ziegler 145

Theological Observer 167
Body, Soul, and Spirit
Proof Text or No Text?
The New Fundamentalism

Book Reviews 175
Icons of Evolution: Science of Myth? By Jonathan Wells.
. . . . Paul A. Zimmerman

(a) (b)

Figure 4.1: Two examples of book highlighting. (a) Content highlighting, copyright of the image belongs to Maryellen Weimer on http://www.facultyfocus.com; (b) Table of contents highlighting, image published on http://backtoluther.blogspot.de, copyright belongs to original author(s)

of these key-points in their lectures and will intentionally or subconsciously emphasize them while teaching [144]. If these emphases can be captured and then presented to students, particularly the self-learning students, it could be very helpful. A good teacher's emphasis should also be the students' learning focus [145].

These thoughts are highly suitable in online learning scenario. As already mentioned in Chapter 1.1, the median engagement time for MOOC learners when watching lecture video is at most 6 minute, but many lectures are much longer than that. So after 6 minutes, learners will become less concentrated or effective, and sometimes even close the video without finishing, causing the phenomenon of "in-video dropout" [146]. However, lecture highlighting may help in this situation.

It is possible to highlight several key sentences in the lecture, which will give learners some kind of sensory impact and work as refreshments to drag learners back from fatigues or distractions. Perhaps these refreshments are not capable to keep learners concentrated all the time, but at least when the key-points are presented, learners are in status of learning. It is also possible to highlight some video segments which cover specific subtopics emphasized by the teacher. With this effort, learners may directly jump to these key segments before "in-video dropout" occurs. In this way, at least learners have already encountered the most important information in this lecture before quitting. If the key segments manage to arouse the interest of some learners, they may even decide not to drop out at all, which is obviously a better outcome.

Meanwhile, the rapid development in video displaying and lecture recording techniques makes the potential implementation of highlighted sentences and segments much easier. In Chapter 2.1, the popularity of enabling subtitle or closed caption while playing video on various platforms has been discussed. These additional textual data, which is synchronized to the video displaying, are very suitable for sentence-level highlight implementation. On the other hand, slide-inclusive lecture recording is also widely adopted, such as tele-TASK and Lecture-Video.NET[1]. By detecting the slide transition, video can be logically segmented and a visual navigation bar can be created, just as introduced in Chapter 3.1 and shown in bottom-right area of Figure 1.1. Key segments can be easily highlighted by simply adding a sign or changing the background color.

Based on all above motives, a solution which highlights the e-lectures in both sentence- and segment-level is proposed in this chapter. The highlighting process is fully automatic by analyzing multimedia course materials. Sentence-level highlighting focuses on acoustic analysis, because a competent teacher should be good as drawing attention from students through voice tone changes when emphasizing key-points in lecture speeches, further to improve the teaching performance [147]. Segment-level highlighting mainly depends on the correlation between speech and slides with statistical analysis, attempting to catch clues when teacher prepares the lecture beforehand.

The rest of this chapter is organized as follow: Chapter 4.2 discusses related works. Chapter 4.3 and 4.4 introduce the sentence-level lecture transcript highlighting and segment-level lecture video highlighting respectively, including approach, implementation and evaluation. Chapter 4.5 compares highlighted key sentences and segments in order to figure out whether there is a connection between them. This is followed by a chapter summary.

[1] http://videolectures.net/

4.2 Related Works

Detecting emphasis in speech is a long term research topic. Early attempts aimed to segment speech recordings or summarize spoken discourses based on the detected acoustic emphasis [148, 149]. From then on, many approaches took pitch as the indispensable feature in this task, since it is widely acknowledged that the pitch value will change as the speaker's status changes [150, 151]. However, there are also some different opinions, such as loudness and duration are more important than pitch in acoustic prominences classification [152]. So in recent years, more approaches prefer to take all pitch, loudness and duration into general consideration.

Syllable-level prominence detection is fundamental in this topic. As the most microscopic linguistic element, the stress of a syllable in a word could be decisive in stress languages like English: "re-***cor***-d" and "***re***-cor-d" can be semantically different. Therefore, there are already many successful systems to automatically classify them in various languages [153, 154, 155]. Then the research interest move upwards from syllable-level to word-level. Acoustically there is no new feature introduced, although discussions have been made about whether to sample on syllables or directly on whole words [156, 157]. Meanwhile, lexical features start to be included in word-level [158, 159].

It is logical to make a similar extension from words to utterances, or we say, sentences. Decent results have been reported in locating "hot spots" of meeting recordings, which is a kind of conversation speech [160, 161]. However, sentence-level emphasis detection on solo speech, to which lecture speech generally belongs, is still limited. It is also a topic to be explored in this chapter.

Once forwarding from sentence-level to segment-level, the information contained in audio signal is no longer enough. Video, with both visual and acoustic content, becomes the major carrier in emphasis analyzing research, and the term "highlight" is more frequently mentioned to address key video segment. Highlight detection is highly well-researched in broadcasting sports videos. The decisive factors include visually detected specific scenes, such as a goal attempt in football/soccer [162] or a home-run in baseball [163], lexically defined keywords in the commentator's speech [164], acoustically measured excitement of the audience

[165], and special replay sessions parsed from professionally produced programs [166]. For other types of video, highlight detection is generally taken as a step in video summarization or abstraction, the video will be deconstructed into shots, on which key-frames will be extracted and further evaluated according to their visual similarity, timing information, and features of synchronized audio signal [167, 168, 169].

However, lecture video is something different. As introduced when discussing segmentation in Chapter 3.2 and 3.5, lecture video has very limited scene changes, which makes almost all the key-frames extracted visually similar. And seldom excitement can be detected from the audience, even when they do exist. But lecture video also has advantages, for example, external multimedia data can be included. He *et al.* made an early attempt with slide-inclusive lectures [170], in which they put audio features, slide transition information and user statistics together into a classification model to extract key segments, although their approach focused more on content integrity than segment importance in purpose of lecture video abstraction. Taskiran *et al.* also contributed to the summarization of lectures [171]. They used pauses in speech to segment the video and calculated importance scores for segments by detecting word co-occurrence based on transcripts. These inspirable ideas arouse a new question: is it possible to find connections between slides and transcripts? This is another topic to be discussed in this chapter.

4.3 Sentence-Level Lecture Transcript Highlighting

4.3.1 Sentence Unit Acquisition

In order to get the sentence units from a lecture video for highlight detection, the most convenient method is to parse the corresponding subtitle file. With the proposed solution of automatic subtitle production introduced in Chapter 2, the process of ***S**entence **B**oundary **D**etection* (SBD) could make lexically logical line breaks in the produced subtitle files, and the subtitle items could be consequently extracted by these line breaks. Since the subtitle file is aligned with the video/audio on the timeline, the time tags of such subtitle items could be applied to cut the audio track. In the end, a sentence unit contains a piece of audio along with corresponding textual content.

Generally, the accuracy of the textual content does not matter so much in following analysis process which is mainly based on acoustic features. But if there are manually created or post-edited subtitles available with better accuracy or even error-free, it is definitely a bonus. Meanwhile, the textual content should be kept in source language during the analysis process.

4.3.2 Voiced/Unvoiced Sound Classification

As mentioned several times, lecture speech is typically solo speech and very likely to be recorded in a professional studio instead of a real classroom. These conditions make the audio signal of lecture videos in high quality with fairly low level of noise. Therefore, a denoising process is not necessary and all acoustic information can be considered as deriving from the speaker. By taking sentence units as input, the first step of emphasis detection is voiced/unvoiced sound classification.

Typically, speech consists of three categories of elements: ***V**oiced sound* (V), ***U**nvoiced sound* (U) and ***S**ilence* (S). For example, the pronunciation of English word "breakfast" should have the structure of "U-V-U-V-U-U", corresponding to "b-rea-k-fa-s-t". And in a sentence of actual speech, "breakfast" would be surrounded by two "S". In many speech analysis tasks, V/U or V/U/S classification is an important pre-processing step [172], emphasis detection is no exception. In this work, voiced and unvoiced sound will be classified by short-term energy and zero-crossing rate.

Short-Term Energy is a basic acoustic feature, commonly used to measure the instantaneous loudness of the audio signal, which will be addressed as 'energy' or E afterwards. ***Z**ero-**C**rossing **R**ate* (ZCR or Z) is the rate of sign-changes along the signal, which can be seen as a simple measurement of frequency within a small time window. It is widely acknowledged that voiced sounds have high energy and low ZCR, while unvoiced sounds are on the opposite: low energy but high ZCR [173, 174, 175]. The silence fragment is easy to classify since both energy and ZCR are approaching 0.

Figure 4.2 shows the energy and ZCR level of an example sentence unit, with the content of "*this is the speed, with which the machine is working*". This

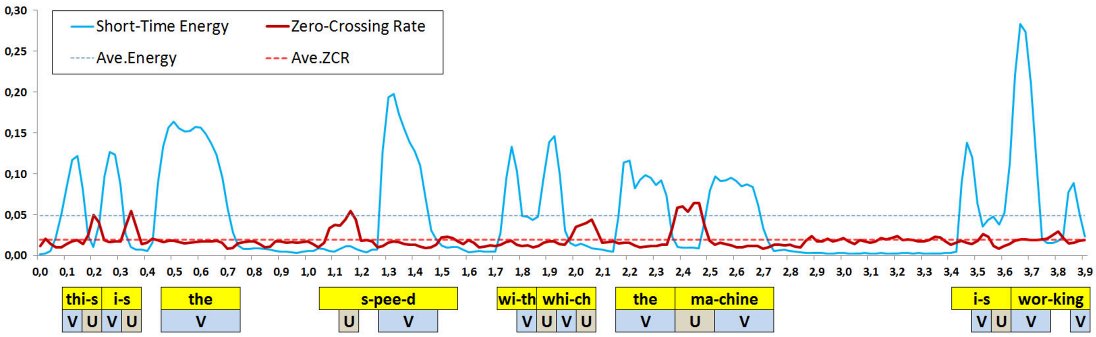


Figure 4.2: The short-term energy and zero-crossing rate of the example sentence unit, along with its voiced/unvoiced deconstruction result.

example comes from the MOOC "In-Memory Data Management" in 2012[1], with the sampling rate of its audio signal as 48 kHz. In this work, both energy and ZCR are sampled with the sampling window size of 0.02 second and the step size of 0.01 second. It means each sample covers 960 sampling points in total and the average value is applied. Both of them are extracted by Yaafe toolkit, just as when preparing the acoustic SBD model in Chapter 2.3.3.

A heuristic-adaptive decision scheme is utilized for the V/U/S classification. First the average energy value of the whole sentence $\bar{E}$ is calculated. For i-th sample in the sentence unit, if $E_i > \bar{E}$ and $E_i > Z_i$, it will be taken as a voiced sample. Then adjacent voiced samples will be connected to form voiced sounds, while independent voiced sample which is not connectable will be considered as accident and abandoned.

Unvoiced sound classification is more complicated. The challenge is to distinguish them from both voiced sounds and noisy silence fragments. After the observation of the speech signals with several different lecturers, following requirements are set:

- ◇ This sample is NOT a voiced sample.
- ◇ $E_i < max\{\bar{E}, Z_i\}$
- ◇ $Z_i > \bar{Z} \times 1.5$ or $E_i + Z_i < \bar{E}$

These requirements demand E_i to be comparatively small, but not too small, and Z_i to be large. An unvoiced sample would be only confirmed when a sample

[1] https://open.hpi.de/courses/imdb2012

satisfies all three requirements. Then continuous unvoiced samples are gathered together just as the voiced samples. After that, all other samples, which are neither involved in any voiced sounds nor unvoiced sounds, will be considered as silence, although some of them might be independent voiced or unvoiced samples.

The result of V/U/S classification on the example sentence unit can also be found in Figure 4.2. Theoretically, there should be 12 voiced sounds and 7 unvoiced sounds based on its textual content. The proposed classification scheme successfully classifies 11 voiced sounds and 6 unvoiced sounds, missing the unvoiced '-d' in "speed", and mistaking voiced sound 'ma-' in "machine" as unvoiced sound. Generally, this scheme can keep the accuracy around 85∼90% when the lecturer speaks calmly and fluently. In purpose of emphasis detection, this accuracy is basically acceptable.

4.3.3 Acoustic Emphasis Analysis

When a speaker attempts to emphasize something, it is natural for him/her to speak louder and/or raise the tone of the voice. Some acoustic features, such as pitch and loudness, will be affected accordingly. Moreover, the speaker definitely wants the audience to clearly capture every single word that is emphasized, for which they might use longer pauses between words to give the audience some extra time for response. In order to catch these possible clues for emphasis detection, following features are measured for analysis:

- **Loudness.** Same as in V/U/S classification, short-term energy is taken as measurement. However, the sampling range is different: only samples involved in voiced sounds are included to calculate and average energy value $\hat{E}$. Each sample in calculation is treated equally, regardless of its position in the voice sound it belongs to or the position of the voiced sound in the sentence unit. $\hat{E}$ represents the loudness level of certain sentence unit.

- **Pitch.** Similar to loudness, the average pitch level will be calculated only within voiced sounds (*addressed as* $\hat{P}$). It is widely believed that males often speak at 65 to 260 Hz, while females speak in 100 to 525 Hz range. But in experiments, the pitch value within the unvoiced sound can easily reach

1000 Hz, which has to be excluded in precaution of interference. Pitch level in this work is extracted by Aubio toolkits.

- **Syllable Duration.** It is reasonable to measure speaking rate by words when the speech sample is long enough. However, since the speech has already been segmented into sentence units with the suggested length shorter than 60 Latin characters (*see Chapter 2.4.2*), the length of a word matters more. For example, the German words "Ja" and "Immatrikulationsbescheinigung", which mean "yes" and "certificate of matriculation" respectively, should not be counted as one word equally. A better measurement is with syllables, where "Ja" has only 1 syllable and "Immatrikulationsbescheinigung" has 11. Ideally, syllables in the transcript should match the voiced sounds in speech one by one [176]. But in practice, the numbers may differ because of the hesitation "eh..." caused by the speaker or the mistake mentioned in Figure 4.2. Therefore, both average syllable duration and average voiced sound duration are calculated, which is also the only place where textual content affects in this approach. Then the smaller value will be applied and addressed as D. D is counted by the number of samples, rather than actual duration of time.

- **Pause Rate.** The pause rate (R_p) is the percentage of silences in the whole sentence unit. It is supposed to be larger when the speaker emphasizes speech elements with extra pauses, as described previously. Practically speaking, the total duration of all classified voiced and unvoiced sounds are summed up and then deducted from the length of the sentence unit. R_p can be considered as an additional feature regarding of speaking rate.

With all above features, an acoustic importance value A_j of the j-th sentence unit in a lecture video with n sentence units in total can be defined as

$$A_j = \hat{E} \times (1 + \frac{j-1}{n-1} \times 0.1) + \hat{P} \times \lambda + D \times \mu + R_p \times \eta \tag{4.1}$$

where λ, μ and η are the weights to balance the influences of different features. They are necessary because the absolute value ranges of the features differ a lot – based on our observation, for instance, $\hat{E} \in [0, 0.5)$ while $\hat{P} \in (100, 500)$.

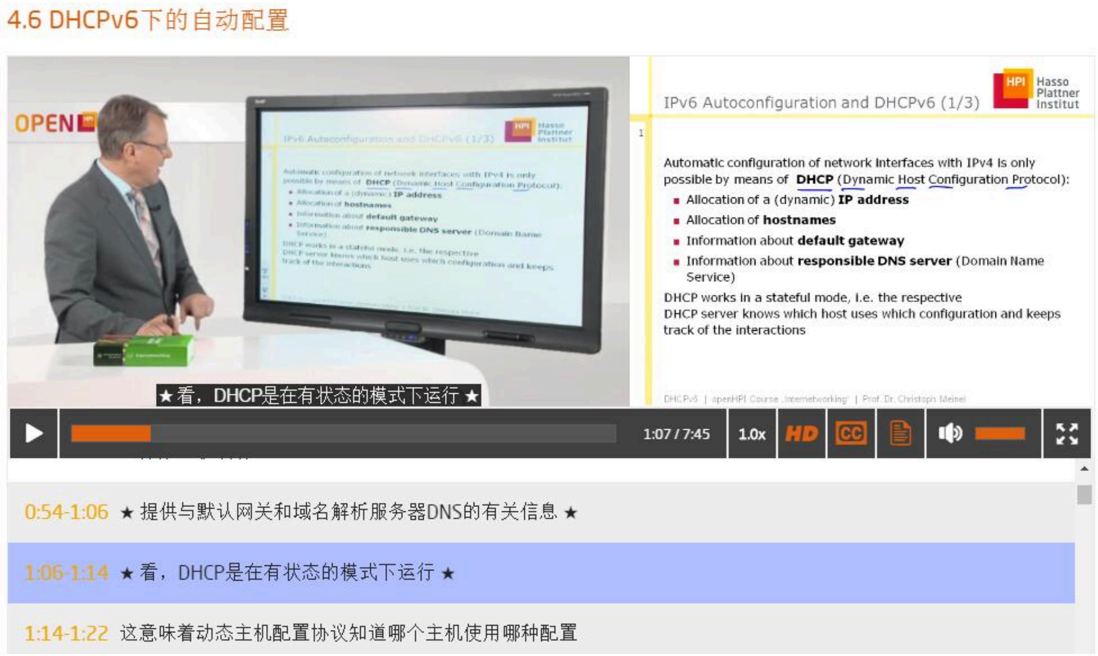
Practically, the average values of the features for each lecture will be calculated and used as benchmark to tune λ, μ and η, in order to make the influences of all four features basically the same. The amendation of $\hat{E}$ is designed because as the lecture proceeds, the speaker will gradually get tired and the loudness level will also decrease unconsciously. The timeline-based amendation could compensate this phenomenon of general energy decay and give a fairer chance to those sentence units in the later phase of the lecture to be detected as emphasis.

The goal of this approach is to highlight a certain proportion of sentence with acoustic emphases from a lecture. Thus there is no need to set a fixed threshold to classify whether a sentence unit is acoustically emphasized or not. A highlighted sentence unit just needs to be more "emphasized" than others. So all sentence units of a lecture will be sorted by their importance values in descending order, among which the top ones will be marked as highlights. Since a sentence unit might not be a complete grammatical sentence, if only one part of the sentence is considered as highlight, so do the other parts. It is the so-called "complete sentence" policy.

4.3.4 Experimental Implementation

After confirming which sentence unit should be highlighted, the next step is to figure out how to present them in a user-friendly appearance when implemented in e-lecture context. It is possible to literally do the "transcript highlighting", by re-formatting the subtitle file into a pure textual transcript file, highlighting those selected sentence units with bold font, background color or underline, just as what people do with traditional paper books, and making it downloadable. However, watching lecture video while checking external reading material simultaneously would not be a pleasant experience for online learners. There is no reason to be optimistic about how it would work.

The highlights need to be implemented in a more easily accessible way, which is better synchronized and embedded with video displaying. Beeping or flashing could be an option, since it is the common way to arouse attention in various scenarios [177]. But in an educational context, they might be too aggressive. Alternatively, it seems the best way is to implement some visual sign for the



4.6 DHCPv6下的自动配置

IPv6 Autoconfiguration and DHCPv6 (1/3)

Automatic configuration of network interfaces with IPv4 is only possible by means of **DHCP** (Dynamic Host Configuration Protocol):

- Allocation of a (dynamic) **IP address**
- Allocation of **hostnames**
- Information about **default gateway**
- Information about **responsible DNS server** (Domain Name Service)

DHCP works in a stateful mode, i.e. the respective DHCP server knows which host uses which configuration and keeps track of the interactions

DHCPv6 | openHPI Course „Internetworking" | Prof. Dr. Christoph Meinel

★看，DHCP是在有状态的模式下运行★

1:07 / 7:45

0:54-1:06 ★ 提供与默认网关和域名解析服务器DNS的有关信息 ★

1:06-1:14 ★ 看，DHCP是在有状态的模式下运行 ★

1:14-1:22 这意味着动态主机配置协议知道哪个主机使用哪种配置

Figure 4.3: The “highlighted” subtitles in MOOC environment, with the sign of star pentagon.

highlighted sentences in the subtitle file. Such signs would make the user instantly aware of these highlights, and some attempts in different context shows its effectiveness [178].

The experimental attempt can be found in Figure 4.3. Each highlighted sentence unit will be surrounded by a pair of star pentagons with solid fill, while a pair of empty star pentagons will be used to mark the previous subtitle item of a highlighted sentence unit as a reminder. The user feedback for this type of implementation will be presented later.

The emphasis detection method applied on sentence-level is generally based on acoustic features, which is language independent. Therefore it should have no problem to run detection on original teaching language but offer result in a translated target language, such as Figure 4.3. It is a screenshot of the MOOC “Internetworking”, which is instructed in English but offered to Chinese-speaking users with subtitles in simplified Chinese[1].

[1] https://openhpi.cn/courses/internetworking2016

4.3.5 Evaluation

4.3.5.1 Methodology

In order to evaluate the performance of the proposed sentence-level highlighting approach, lecture 4.5 and 4.6 of “Internetworking” are used as test data. It is a comparatively limited scale evaluation, which consists of three aspects. Firstly a few highlighted examples are demonstrated, with explanations of the rationality behind them. Then the precision is calculated based on the ground-truth created by multiple experts. Finally the user feedbacks are presented. In this experiment, the coefficients in equation (4.1) are set as follow: $\lambda = 0.001$, $\mu = 0.02$, $\eta = 1$, according to the principle introduced in chapter 4.3.3, and the selection proportion is around 1/6.

4.3.5.2 Example Demonstration

The first example is a fraction of lecture 4.5 which talks about ***N**eighbor **D**iscovery **P**rotocol* (NDP). The total length is 7:28 and it is segmented into 58 sentence units in the subtitle file. Among them 10 units are highlighted based on the proposed acoustic emphasis detection, plus 2 added by the “complete sentence” policy. Here the pure textual content are extracted from the subtitle file for better presentation, with the highlighted part marked by bold font. The first example is:

> *“. . . And the first, I want to mention is the neighbor discovery protocol. The task of the neighbor discovery protocol, NDP, is to facilitate the interaction between adjacent nodes. What are the adjacent nodes?* ***They are neighbor nodes. In IPv6 nodes are considered adjacent, if they are located on the same ‘link’.*** *And the IPv6 link is the network area that is bounded by a router . . . ”*

Grammatically, the highlighted content in this example is the answer of a “hypophora”, or addressed as “anthypophora”. In plain language, hypophora is a self-answering question, which is generally believed to be used to draw attention or arouse curiosity from the audience [179, 180], in order to further heighten the effect of what is being spoken, in other words, to create emphasis. This phenomenon exists widely in educational context [181, 182]. Semantically, the highlighted content is the explanation of an important technical term – neighbor node – in a lecture talking about NDP. All these facts strongly suggest that highlighting this sentence is highly logical.

As mentioned before, “Internetworking” is a MOOC instructed in English but offered in Chinese. The subtitle prepared is completely in simplified Chinese and above example is actually presented as:

> “...首先我要提的是邻机发现协议。邻机发现协议NDP的任务是协助邻近节点之间的互动，什么是邻近节点？**这些是邻近节点。在IPv6中如果节点位于相同的“链路”，则它们被认为是邻近节点。**IPv6链路是指一台路由器覆盖的网络区域...”

Here the Chinese text is quoted because, due to the consideration of word order and fluency, the second and the third highlighted sentence units reverse their positions when translated from English to Chinese (*for non-Chinese speakers, please focus on the different positions of the quotation marks in the example*). However, it is not influenced since the “complete sentence” policy is applied.

The second example derives from lecture 4.6, which talks about the *__D__ynamic __H__ost __C__onfiguration __P__rotocol* (DHCP) under the framework of IPv6. This lecture lasts for 7:45, with 66 sentence units in total, 12 of them acoustically highlighted and 1 added for “complete sentence”. This example actually corresponds to Figure 4.3:

> *“...In IPv4, this was only possible with the DHCP protocol, the Dynamic Host Configuration Protocol.* ***The DHCP protocol was responsible to dynamically allocate IP address to the host, to allocate the host names, to provide information about default gateway, and information about responsible DNS server (Domain name service). See DHCP protocol works in a stateful mode.*** *That means the respective DHCP server knows which host uses which configuration and keeps track of all the interactions ...”*

Table 4.1: Precision analysis on sentence-level highlighting

Video	All Sentences		GT-Key	Highlighted Sentences			
	Num	Ave		Num	Ave	Hit	Precision
L4.5	58	0.78	18	10	1.12	7	70.0%
L4.6	66	0.93	20	12	1.16	7	58.3%
All	124	0.86	38	22	1.14	14	**63.6%**

By comparing with the slide recorded in the right section of Figure 4.3, it is clear that the highlighted part in this example is the same as what is written in the slide. People use slides as the outline of the talk, in other words, the slide is the collection of important terms the speaker wants to mention. Detecting them as key sentences seems to be a good decision.

Unfortunately, not all detected highlights can be supported by strong theoretical explanations as the previous examples. Therefore, a quantitative precision analysis is necessary.

4.3.5.3 Precision Analysis

In this subsection, the general precision of highlighted sentences will be evaluated in a fairly objective way. Unlike typical classification questions, it is impossible to have the absolute objective ground-truth in this context, because whether a sentence should be highlighted or not is a subjective question, the answer may differ from different people. However, if several people, especially several experts in related topics, share similar opinion, it could be taken as reference. Then the reference can be further considered as ground-truth.

Still, lecture 4.5 and 4.6 of “Internetworking” are used as testing data. For each lecture, 5 experts are invited to rate the importance of each sentence unit with three levels: 2 (*recommend as highlight*), 1 (*neutral*) and 0 (*not important*). All these experts graduate from IT-related majors in different universities, still work in this profession and are familiar with the topic of NDP or DHCP of the testing lectures. After receiving the ratings from these experts, an average value would be calculated for each sentence unit. If the value of a sentence unit is greater than 1, it will be marked as a ground-truth key sentence. In “GT-Key” column of Table 4.1, the number of such key sentences can be found.

Please note that in precision analysis, “complete sentence” policy is not applied. As can be seen in Table 4.1, the ground-truth key sentences are considerably more than acoustically highlighted sentences. In such condition, it is less meaningful to evaluate the recall rate, so only precision is reported. “Ave” represents the average importance value rated by experts within certain group of sentence units, while “Hit” means the number of sentence units which are both highlighted by proposed approach and recommended by experts. Result shows that 14 of 22 highlighted sentence units are “correct”, with the precision as 63.6%.

4.3.5.4 User Feedback

Besides the subjective and objective evaluation from developer’s side, opinions directly from user’s side are also crucial. In MOOCs only when a new feature is welcomed and used by learners, may it actually be beneficial. Therefore a survey is prepared to observe the general acceptance of proposed sentence-level approach in form of highlighted subtitles. Since all survey items are optional in principle and independent from each other, the total number of replies per item could be different.

When talking about the prospect of newly developed techniques, **T**echnology **A**cceptance **M**odel (TAM) is frequently referenced [183], in which “perceived usefulness” and “perceived ease-of-use” are considered as the basic reactions for the users who encounter new technical stuff, and then form the “attitude towards using” and finally affect actual use. As illustrated in Table 4.2, 76.7% of survey respondents acknowledge the positive meaning of proposed new feature (Q1), while 77.8% did notice the existence of highlighted sentences (Q2). These numbers prove the potential usefulness and the convenience of proposed work.

Regarding technical details, users expressed different and somehow contradictory opinions about the way how lecture highlights are implemented (Q3), and many of them are basically satisfactory with current accuracy (Q4): an average rate of 3.66 is achieved and can be transformed into 66.5% in percentage, which is similar to the objective precision obtained (63.6%). Finally, 76.5% of users explicitly indicated their “Yes” attitude towards this new feature by supporting the formal adoption of subtitles with highlighted items in follow-up lectures and courses, while another 13.2% are not against this idea either (Q5).

Table 4.2: Statistics about the Survey

Q1: Do you think that we offer sentence-level highlights in lectures is meaningful in context of MOOC?	Count	Ratio
(1) Yes	56	76.7%
(2) No	6	8.2%
(3) I'm not sure	11	15.1%
Q2: Have you noticed the highlighted sentences in previous lectures?	Count	Ratio
(1) Yes	56	77.8%
(2) No	16	22.2%
Q3: Do you think our current implementation, with star polygon pairs, is appropriate?	Count	Ratio
(1) Yes, it's completely appropriate.	32	47.1%
(2) It's OK, but the reminder is unecessary.	14	20.6%
(3) It's OK, but the sign should be more obvious.	13	19.1%
(4) It's OK, but the sign is too garish.	5	7.4%
(5) No, it's terrible.	4	5.9%
Q4: Please rate the current accuracy of the highlights offered. (5-star is the highest and 1 is the lowest)	Count	Ratio
(1) ⋆⋆⋆⋆⋆	20	28.2%
(2) ⋆⋆⋆⋆	22	31.0%
(3) ⋆⋆⋆	21	29.6%
(4) ⋆⋆	1	1.4%
(5) ⋆	7	9.9%
Q5: With current level of accuracy, do you want us to formally apply highlighted subtitles in following lectures and courses?	Count	Ratio
(1) Yes	52	76.5%
(2) No	7	10.3%
(3) I'm not sure	9	13.2%

Generally Speaking, the user feedback is positive. However, it cannot be neglected that the scale of this survey is quite limited, not only because of the comparatively small number of total participants, but also because of their homogenization – all respondents are native Chinese speakers. Moreover, the structure of above survey and the methodology of feedback analysis are also relatively simple. Definitely, it would be better to get more ideas from a larger and more diverse user group.

4.4 Segment-Level Lecture Video Highlighting

4.4.1 Segment Units Preparation

In segment-level lecture highlighting, the first task is to define segment. Lecture video segmentation has been discussed in Chapter 3, where slide transitions are detected and utilized to split the lecture video into fragments, and then logical correlation between adjacent fragments are explored in order to gather such fragments into groups, which further become segments with average length of 7:36. However, these segments are perhaps too long to be considered as the basic unit for highlighting, although this method can still be applied to split those extra-long lectures into several independent sub-lectures. In purpose of highlighting, a lecture or sub-lecture will be again split into slide-transition-based fragments. These lecture fragments will be addressed as SUs (***S****lide* ***U****nits*) and taken as the basic unit in segment-level lecture highlighting.

Obviously, SUs are only available with slide-inclusive lectures, just as the discussion in Chapter 3. For each SU, a textual outline can be generated by executing OCR on the corresponding screenshotted slide image or directly by parsing external slide file, with the proposed TOG approach. On the other hand, based on the beginning and ending time tags of SUs, subtitle files can also be split and each SU will contain a paragraph of the lecture transcript. No matter the subtitle files are generated automatically by the approach described in Chapter 2 or created manually, they can be compared and analyzed with corresponding sections of outline, which is also the basic idea behind proposed segment-level highlighting approach.

Based on the outline and transcript, each SU will have following direct parameters:

- ◇ Type: *T-SU* (pure textual slide), *NT-SU* (except for the title, there is no text in the slide but only illustrations, such as chart, image, etc.) and *HT-SU* (mixed).
- ◇ Duration (d): counted in second.
- ◇ O-Words (W_O): total number of words in the slide outline.
- ◇ O-Items (I): total number of textual items in the slide outline, including title, topics and subtopics.
- ◇ S-Words (W_S): total number of words in speech paragraph.
- ◇ Co-Occur (C): total number of words shared by both slide outline and speech paragraph.

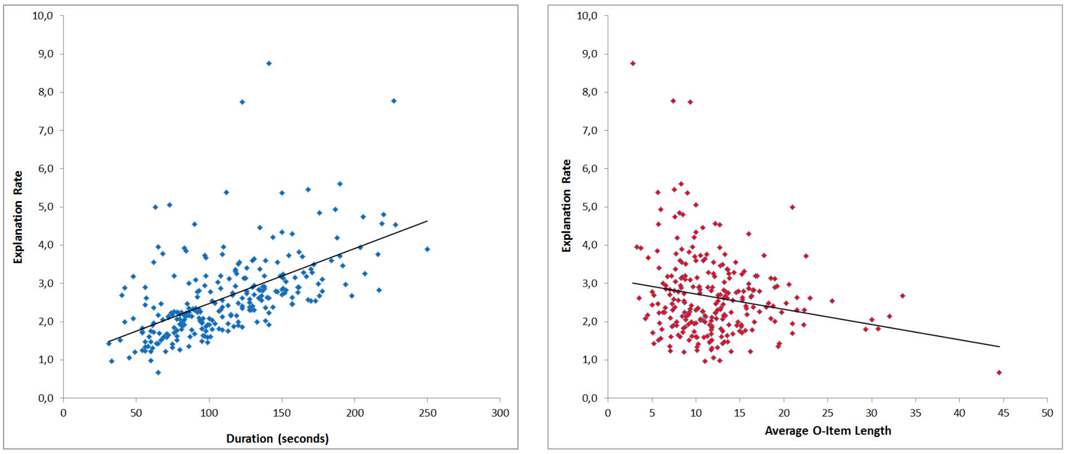
Based on these direct parameters, several indirect parameters are defined to better represent the characteristic of the SUs, which include:

- ◇ Speaking Rate: $R_S = W_S/(d/60)$
- ◇ Matching Rate: $R_M = C/W_O$
- ◇ Explanation Rate: $R_E = W_S/W_O$
- ◇ Average O-Item Length: $L_I = W_O/I$
- ◇ Average O-Item Duration: $d_I = d/I$

Since not all above parameters are available or meaningful for all types of SUs, the following analyzing process will also be categorized based on the SU types.

4.4.2 Importance Analysis of T-SU

For educational purposes, slides generally serve as the outline of the textbook. In many slides, the lecturer lists the subtopics one by one, in addition to some short explanations, making the slide as a collection of textual content. This type of SU is T-SU. However, the lecturer should always offer some extra information in the lecture speech, otherwise learners could simply read the slides by themselves. The importance analysis of T-SU would mainly focus on exploring the connection



(a) Ascending trend while d increases.

(b) Descending trend while L_I increases.

Figure 4.4: Course-scale R_E distribution with certain parameters.

between the information in speech and slide. This would involve the following factors:

1) Expected Explanation Rate. The idea here comes from a simple assumption: the lecturer will explain in more details when talking about something important. Naturally, a T-SU in such conditions will have a comparatively higher explanation rate R_E. However, taking the absolute value of R_E as the measurement might not be suitable, because when collecting data from a complete course (*MOOC "Web Technologies"*), there is an apparent ascending trend of R_E with the increase of SU duration. Figure 4.4-a illustrates this trend clearly. So the concept of "expected explanation rate" is introduced, which is estimated by the lineal trend line fitted in Figure 4.4-a. Since the value is related to the SU duration d, it is addressed as $\tilde{R}_{E(d)}$.

Similarly, a second expected explanation rate could be estimated based on the course-scale observation of another SU parameter: average item length L_I. Smaller L_I refers to the existences of more keywords or keyphrases in the slide, while larger L_I indicates there might be more complete sentences. Then it is quite understandable that a lecturer needs to add more extra information in the speech when L_I is decreasing. Figure 4.4-b captures this trend with the descending trend line, by which $\tilde{R}_{E(L_I)}$ can be calculated.

Then the difference between the expected explanation rates and the actual value can be taken as a measurement. The first evaluation factor of T-SU, f_E, would be calculated by:

$$f_E = R_E - \frac{\tilde{R}_{E(d)} + \tilde{R}_{E(L_I)}}{2} \tag{4.2}$$

where f_E might be either positive or negative, and we expect the value of f_E could be large and positive if the content of relevant T-SU is important.

2) Hypothesis on Speaking Rate and Matching Rate. As already mentioned several times, lecture speech is generally solo speech and recorded in studio with very limited interference. In such scenarios, the lecturer is uninterrupted and easy to keep the speaking rate R_S stable. However, experienced teachers know that a lecture should not be given like a lullaby. When, where and how to make emphasis is very important to improve teaching quality, by which intentional slowing down is a frequently used and effective trick [184, 185].

But it is inappropriate to directly take low speaking rate as evidence of emphasizing. Pause from hesitation may also result in low speaking rate [186]. Unfortunately, it is very difficult to avoid, even for experienced teachers. In this case, to distinguish whether a slow-down event is intentionally or accidentally caused just by simply checking the speaking rate is not enough.

A hypothesis is made on when the lecturer is intentionally slowing down. Matching rate R_M is introduced here to evaluate the degree of overlapping in the texts of speech and slide. Then based on different conditions of R_S and R_M, four scenarios are visualized:

- ◇ High R_S, Low R_M: the lecturer is introducing something less related, such as a background story or homework requirement. The content is not written in the slides but the lecturer is familiar with it.
- ◇ Low R_S, Low R_M: the lecturer is talking about something unprepared. Unfamiliarity causes hesitation and pauses in speech, while the content has not been written in the slides.
- ◇ High R_S, High R_M: the lecturer is reading the slides in high speed.
- ◇ Low R_S, High R_M: the lecturer is making an intentional emphasis by slowing down.

Based on this hypothesis, low R_S and high R_M is the desired condition and a second evaluation factor f_H for T-SU can be illustrated as:

$$f_H = \frac{(R_M - \bar{R}_M) \times 100 - (R_S - \bar{R}_S)}{2} \tag{4.3}$$

where $\bar{R}_M$ and $\bar{R}_S$ refer to the average values of R_M and R_S of the whole course. Important T-SUs are supposed to possess larger and positive values of f_H.

3) Overview Bonus. Many MOOCs are designed for the purpose of popularizing science. A certain lecture in one of such courses is very likely to be an initial introduction about a specific topic, not academically advanced, but covering as many subtopics as possible. For example, if a lecture is about a first glance of programming languages, it may introduce C, C++, Java, Python, *etc.*, separately and briefly. For these lectures, overview is the most important part and there is probably an overview slide placed at the beginning of the corresponding video. In this approach if a video clip is not long (*less than 10 minutes*), contains only few slide pages (*less than 10 pages*) and the first slide is an independent slide, which means it cannot be combined with following slides according to the scheme introduced in Chapter 3.4.5, then it will be acknowledged as an overview page and earn a bonus (B_O).

By now all three factors can be summarized into a final importance value of T-SU: $V_T = f_E + \lambda \times f_H + B_O + \mu$, where λ is a weight to adjust the influence of f_H and μ is a course-based fixed offset to make V_T always positive. Certainly the value of V_T is supposed to be larger in key T-SUs.

4.4.3 Importance Analyses of NT-SU and HT-SU

In a NT-SU, the slide structure is quite simple: a title and a full-page illustration. It might be a chart, a diagram, an image or specifically in IT-related lectures, a code block. Since O-Words (W_O) is meaningless in such slides, several features used in T-SU are no longer available, such as the explanation rate and the matching rate. Thus only a simple measurement is adopted here: the total amount of information contained by a NT-SU, which depends on the S-Words (W_S). It is supposed that if a full-page illustration is introduced for a key procedure in a technique or a significant exhibition of an important system, the lecturer would explain more in the speech, with a large W_S as a result. And for those illustrations which the lecturer just briefly mentions in a few words, it could not be the highlight. So the importance value of NT-SU is defined as: $V_{NT} = W_S$.

The situation of HT-SU is in between of T-SU and NT-SU. With illustrations occupying half the page, there is still a considerable portion of text, which makes all SU parameters available. But it is very difficult to quantify the proportion of information carried by text and illustrations, thus explanation rate becomes much less convincing in comparison with T-SU. Alternatively, the average item duration d_I is implemented as the measurement of how detailed the lecturer teaches within the certain HT-SU. Each illustration would be counted as an additional item in the slide.

On the other hand, similar to NT-SU, the importance of a HT-SU might also be positively related with the amount of information the lecturer offers, including both W_S and W_O. The importance value of HT-SU will be set to

$$V_{HT} = \frac{W_S + W_O - C}{2} + d_I \tag{4.4}$$

where C is the co-occurrence, which we intend to remove as redundancy. V_{HT} shall be large when the HT-SU is a key segment.

4.4.4 Ground-Truth Acquisition

In order to evaluate whether the highlighted segments by proposed approach are correct, survey questions are offered in self-tests of the MOOC “Web Technologies”, which is a 6-week course instructed in English on openHPI.de platform[1]. 10022 learners enrolled in this course during the opening time, 1328 participants took the final exam and 1179 of them successfully earned the certificates.

The survey covers 43 lecture video clips with a total length of 632 minutes, in which 348 SUs are automatically obtained. In the survey learners were asked to select one segment as the most important one in the correlated lecture video which they just finished watching. Over 5000 replies were received for the first video, and as users dropping out, there were still over 1000 users who took part in the survey of last video.

[1] https://open.hpi.de/courses/webtech2015

For a video with n SUs in total, if u_i users choose the i-th SU as the most important one, its basic importance factor IF_i will be set to

$$IF_i = \frac{u_i}{\sum_{j=1}^{n} u_j} \times n \tag{4.5}$$

By this calculation, the importance factor can better represent the extent how important the corresponding SU is in users' point of view. It bases on the proportion of users who select certain SU as most important, not the absolute number, which could avoid the negative influence of varying numbers of survey participants. It is also related to the total number of SUs in the videos. Obviously earning 33% of votes in a 10-SUs video is already high enough for a SU, but earning 33% in a 3-SUs video is just on the average. This feature is also well represented in (4.5). Mathematically, the average value of IF_i in either a lecture or the whole course is 1.

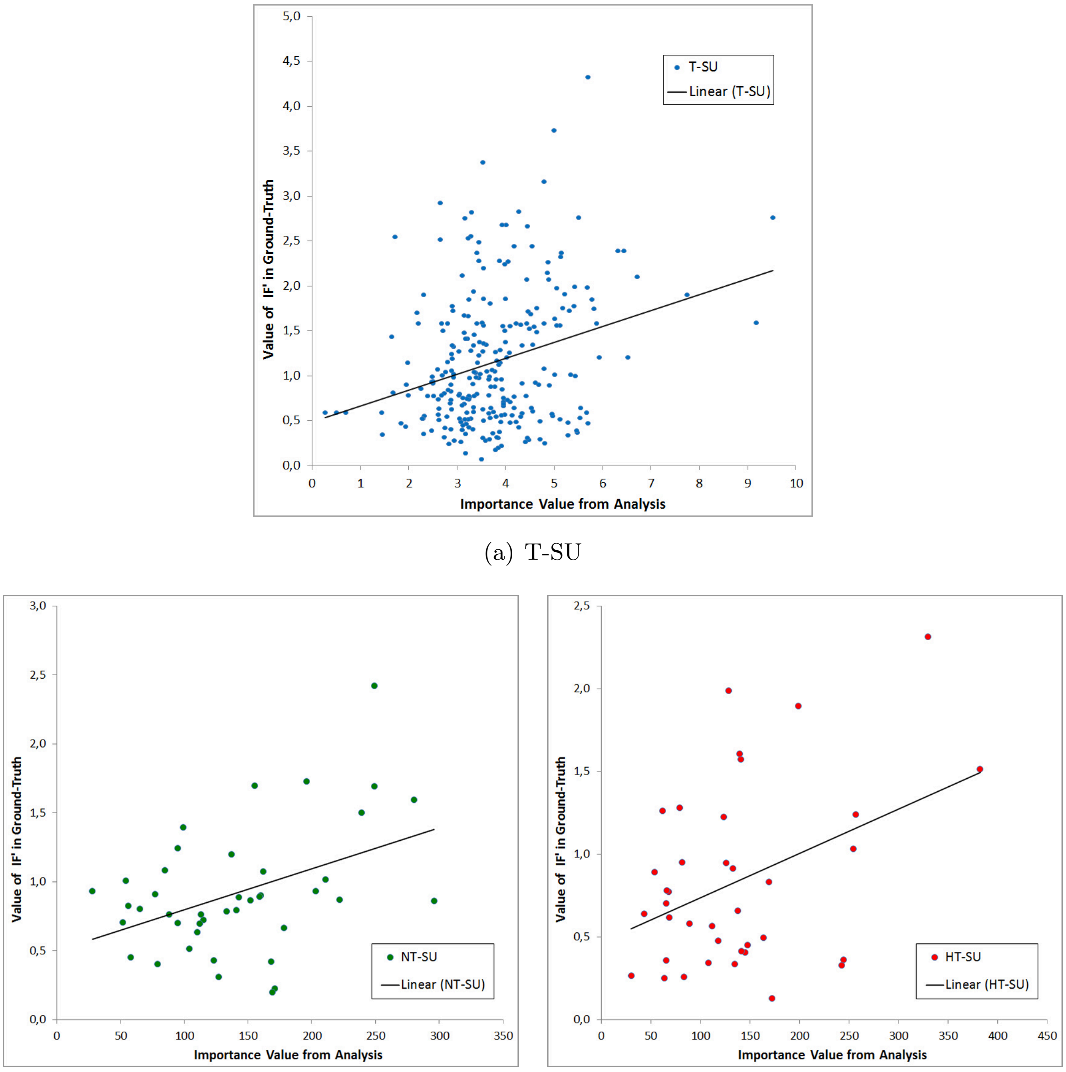
Moreover, the course-attached discussion forum could also offer valuable user feedbacks. Generally, only important content of the lecture would intrigue learners to ask questions. Based on content, forum threads are assigned to SUs, and each thread earns a small bonus for the importance factor of related SU. This bonus is also balanced since there are obviously more questions in forum in the early stage of the course than the later stage. If the i-th SU has q_i related questions, the final importance factor IF'_i is set to

$$IF'_i = IF_i + \frac{q_i}{\sqrt{\sum_{j=1}^{n} u_j}} \times \eta \tag{4.6}$$

where η is a coefficient to keep the bonus value proper and can only be set manually based on how many forum threads are created. For "Web Technologies", η is set to 10, which makes each question worth 0.1~0.2. Since this coefficient is only valid in evaluation, it will not affect the automatic process of detecting highlights. In the end, IF'_i will be taken as the ground-truth for i-th SU in following evaluation.

4.4.5 Evaluation

Based on the data collected from "Web Technologies", the detailed coefficients are set as follow: $\lambda = 0.1$ to keep the influences of f_E and f_H on same level, $\mu = 3.5$



(a) T-SU

(b) NT-SU

(c) HT-SU

Figure 4.5: General trend illustration for all 3 types of SUs.

to shift V_T beyond zero. Since V_T, V_{NT} and V_{HT} have different definitions, the evaluation result would be first shown separately in Figure 4.5. The calculated importance value (V_T, V_{NT} or V_{HT}) is the variable in x-axis while the ground-truth importance IF' is in y-axis. By the trend lines fitted, it is easy to figure out that no matter for T-SU, NT-SU or HT-SU, a positive relation between two variables is clear.

More specifically for T-SUs, which are the majority in all SUs, the positive

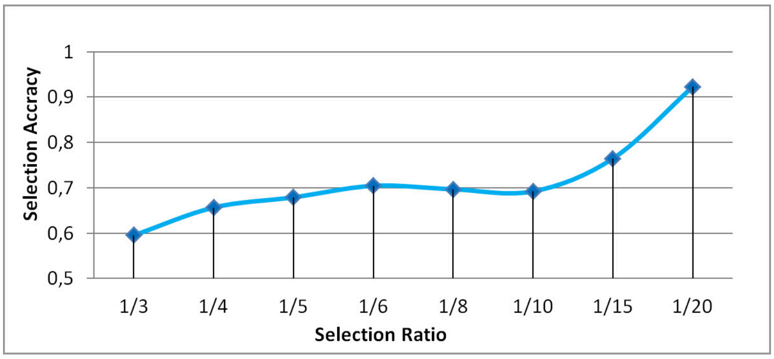


Figure 4.6: Precisions with different selection rate. A ratio of $1/k$ means when sorting all SUs with the calculated importance descending, top $1/k$ SUs will be selected as "highlighted segments"

result is fundamentally contributed by f_E, with explanation rate as the core factor. The hypothesis about matching rate and speaking rate (f_H) does not meet the initial expectation. The overview bonus B_O, although it can only affect a small portion of T-SUs, is proven to be very helpful.

Except for the general trend, how to facilitate potential application should also be considered. Similar to sentence-level highlighting, there is no need to use an absolute threshold to classify whether a segment or SU should be highlighted, while the ranking of segments or SUs matter more. If all SUs are sorted in descending order of their calculated importance values and a certain portion are selected from the top, these selected segments can be further evaluated by precision before finally offered to learners. The correctness is defined as follow: if a selected segment has a ground-truth IF' greater than 1, then it is correct.

Again, three types of SUs will be evaluated separately. T-SU is taken as example in Figure 4.6 to see the precision change as the selection ratio decreasing, based on the data of "Web Technologies". As can be seen, 1/6 is the optimized option which can balance the quantity and quality of selected segments. It is also the selection ratio of sentence-level highlighting.

When 1/6 is also applied as selection ratio for NT-SU and HT-SU, more statistics is listed in Table 4.3. The precisions for selected segments are 70.5%, 71.4% and 66.7% for three types of SUs respectively. "Ave-IF'" represents the average IF' of related SU group. It is obvious that the ave-IF' of highlighted

Table 4.3: Precision analysis on top 1/6 Selection

Type	All Segments		Highlighted Segments (*Top 1/6*)			
	Num	Ave-IF'	Num	Ave-IF'	Correct	Precision
T-SU	268	1.15	44	1.58	31	70.5%
NT-SU	42	0.82	7	1.40	5	71.4%
HT-SU	38	0.83	6	1.13	4	66.7%
All	348	-	57	-	40	**70.2%**

segments is higher than the average of all. The general precision, 70.2%, is fairly promising.

4.5 Potential Connection between Highlighted Sentences and Segments

Since the results from both sentence-level and segment-level lecture highlighting are positive (*63.6% and 70.2%*), it is quite natural to ask, is there any connection between the highlighted sentences and segments? In order to figure it out, one additional experiment is introduced, in which the sentence-level acoustic analysis is applied on the testing data previously used in segment-level statistical analysis, the MOOC "Web Technologies". The sentences units are distributed to the slide units based on the time tags, averagely 18 to 1, and the following features are collected per slide unit (SU):

- **The normalized mean acoustic importance value.** Firstly the acoustic importance value is achieved for every sentence unit and then a simple mean value (A_i *for the i-th SU in the lecture*) is calculated, with each sentence unit with same weight. However, A_i is only comparable within the specific video, because the absolute value of this acoustic-based feature varies too much in different videos, due to recording conditions and lecturer status. Therefore, a lecture-average acoustic importance value ($\bar{A}$) is further calculated, and a normalized $\hat{A}_i$ is defined as: $\hat{A}_i = A_i/\bar{A}$.
- **The standard deviation of acoustic importance value.** The calculation is carried out exactly according to the mathematical definition of stan-

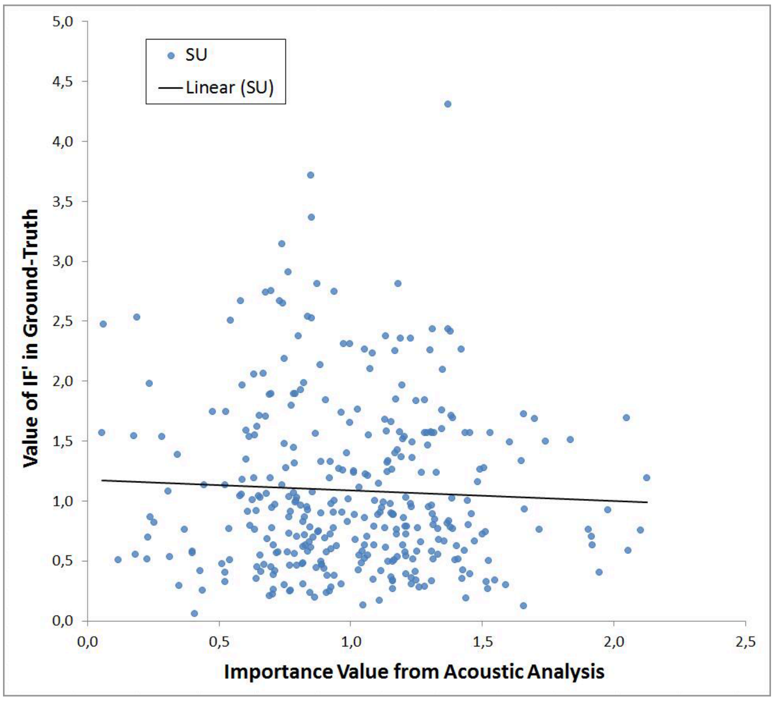


Figure 4.7: The distribution of data points and simulated linear trend line of acoustic importance value and ground-truth importance factor.

dard deviation, working on all sentence units inside a SU. It is addressed as D_i

- **Highlighting Rate.** A simple ratio of the highlighted sentence units in all sentence units, which is addressed as R_i.

If there is a positive relation between key segments and key sentences, a highlighted segment should be supported by more highlighted sentence units, which means larger values of $\hat{A}_i$ and R_i. Meanwhile, emphasis needs comparison between peak and trough, which is supposed to result in a larger D_i. Again some static coefficients are used to balance the influences of these features and sum them up by $V_A = \hat{A}_i + \lambda \times D_i + R_i + \mu$, and then project the sum in Figure 4.7 against the ground-truth importance factor IF', with $\lambda = 10$ and $\mu = -1.5$. However, the data points in Figure 4.7 are randomly distributed and no ascending trend line can be fitted. It means, based on the data of "Web Technologies", there is no evidence to support the theory that highlighted segments are constructed by highlighted sentences.

Although this result is not ideal, it is actually logical. Sentence-level highlights in proposed approach are in fact based on acoustic prominences. No matter whether it concerns the loudness, tone or speaking rate, the prominence is a short-time phenomenon and focuses only on local context. The lecturer is indeed attempting to make an emphasis intentionally, but the physiological response which finally creates the prominences is subconscious.

A segment, however, averagely consists of 18 sentence units and lasts for 109 seconds in “Web Technologies”. During such a long period of time, a speaker could not consistently offer acoustic prominences. Otherwise he/she would be sound over excited and quickly getting tired. An experienced teacher should avoid such behavior when giving a lecture.

Therefore, it is understandable that the key segments are not acoustically significant and consequently not having a positive relation with key sentences. Segments are highlighted in this approach mainly depending on high explanation rate, large information amount and overview bonus, all of which are structural elements and originate from thoughtful decisions when the teacher prepares the lecture beforehand.

As introduced in Chapter 4.1, although both sentence- and segment-level lecture highlighting aim to improve online learning experiences for learners, the detailed goals are different. Sentence highlighting in subtitles works as a reminder to keep learners focused. Segment highlighting is more like a selector, which gives learners a better navigation. In this point of view, the acoustic-based key sentences and the structural-related key segments could accomplish their tasks separately. There is no need to force seeking a connection between the two.

4.6 Chapter Summary

In this chapter, two different approaches are proposed to extract highlights from online lectures. The first approach is mainly based on acoustic analysis, which would output highlights in sentence-level. The second approach focuses on statistical features, analyzing which segment of the lecture video should be highlighted by exploring correlation between slides and speeches. Then an attempt is made to see whether there is a connection between highlighted sentences and segments.

Sentence-level highlighting starts with voiced/unvoiced sound classification. Then the importance of a sentence unit will be analyzed based on the voiced and unvoiced sounds classified. Short-term energy, zero-crossing rate, pitch and some other features are involved. After experimentally implementing into the subtitle files of several online lectures, the highlighted sentences are evaluated with example demonstration, precision analysis and user feedbacks with fairly good result.

Segment-level highlighting is only available for slide-inclusive lectures. The lecture video is split into slide units based on slide transition detection. Then both the slide content and the speech transcripts are extracted for these units. Structural features like explanation rate, matching rate, speaking rate, *etc.*, are considered in importance analysis. After collecting user feedback from the MOOC “Web Technologies”, the proposed segment-level highlighting method is evaluated in whole course scale and shows quite promising result.

Using same data of “Web Technologies”, an attempt to find potential connection between highlighted sentences and segments is made, but not successful. The possible reason is also discussed. However, both sentence- and segment-level highlights could help online learners in their own way and technically, there is no obstacle to implement them into e-lecture or MOOC context.

Chapter 5

Conclusion

As the rapid development of distance learning technology, more and more e-lectures are recorded professionally and uploaded for public access. However, no matter in traditional tele-teaching or in recently popular MOOCs, lecture video is always the core material. In this thesis, several technical solutions are proposed to enhance the learning materials other than videos, in order to further facilitate online learners. These solutions work automatically based on multimedia analysis, without giving extra burden to teacher.

The first solution is *Lecture Subtitle Production.* Beside of implementing reputable ASR and MT tools, a novel SBD approach is proposed in between. This SBD approach consists of an efficient lexical model based on DNN and WV, a simple but capable pause-only acoustic model as well as a highly flexible 2-stage joint decision scheme, whose performances reach state-of-the-art. Furthermore, by training a set of German WVs, SBD service has been successfully extended from English to German, and similar method can be applied in other languages.

With the better segmenting positions and punctuation marks restored by SBD, the quality of auto-generated subtitles is apparently improved and better than the average of manually created subtitles by dozens of volunteers from scratch. But a more significant contribution of such auto-generated subtitle is that by taking them as draft, human post-editor could save over 1/3 of total working time with no quality decline, especially with source language. The solution of automatic *Lecture Subtitle Production* has already been applied in preparation of several MOOCs and is highly appreciated by corresponding teaching teams.

The second solution is *Lecture Outline Generation*, by which TOG is proposed to extract the textual data from lecture slides recorded in videos. With STD and OCR as pre-processing, TOG first reconstructs the intra-slide content system by page layout analysis, then explores inter-slides logic to locate subtopics and finally achieves an up-to-6-level tree-structure lecture outline. The general accuracy of this outline is 85%, which assures its functionalities as preview, navigation and retrieval. Moreover, table detection and lecture video segmentation are taken as sub- and post-applications of TOG, which also achieve very encouraging result.

Based on the subtitles and outlines previously created, *Lecture Highlighting* is further conducted. It works in both sentence- and segment-level. In sentence-level highlighting, acoustic emphases are classified from the lecture speech, based on the analysis of features like short-term energy, ZCR, pitch, *etc.* Highlighted sentences are marked in lecture subtitles and both objective and subjective evaluations prove its rationality. On the other hand, segment-level highlighting focuses on statistically analyzing the correlation between lecture speech and slides. Structural features like explanation rate, speaking rate and matching rate are taken into consideration. By evaluating with ground-truth generated by massive users, the precision of highlighted segments are also highly promising.

With the subtitles, outlines and highlights provided, online learning is no longer just staring at the lecture videos. Learners could understand those lectures instructed in different languages, find wanted lectures more precisely and better arrange their learning schedules. Since all these enhanced learning materials are obtained from teacher's side, the whole preparation can be done pre-course, which means the proposed advantages are not only available for archived e-lectures, but also applicable for on-going courses, particularly the MOOCs.

In the future, it is definitely necessary to further improve the quality of auto-generated subtitles, outlines and highlights. Especially for the subtitles in target language, how to achieve better MT output is a very challenging task. Besides, it is also possible to build interactive user interfaces and invite e-learning portal users to help modify these learning materials, enhancing the learning experiences from the student's side. It's never too late to learn, and it's also never too late to improve.

Appendix A

Settings of Some Parameters

There are some detailed parameter settings which have not been illustrated in main text. They can be found here. Table A.1 introduces the parameters set in Chapter 3.3.2.2.

Table A.1: Some weighted values in "Table Candidate Evaluation"

Content Mark		Column Bonus		Distance Deduction	
Type	Weight	Type	Weight	Condition	Weight
Digit	+2.5	Digit	+1.01	gap < width/2	+0.2
Single Word	+1	Word	+0.5	gap < width	-0.2
Length $\leq$ 10	+0.5			gap < width × 1.5	-0.2
10 < Length < 25	0			gap < width × 1.75	-1
25 $\leq$ Length < 40	-0.5			gap < width × 2	-2
Length $\geq$ 40	-0.5			gap < width × 3	-3

Table A.2 introduces parameters used in Chapter 3.3.3, where G_{max} means the maximum gap either horizontally or vertically.

Table A.2: The threshold to include new Text-lines in "Table Area Expansion"

Target	Horizontal	Vertical
Digit	$2.5 \times G_{max}$	$2.5 \times G_{max}$
Single Word	$1.875 \times G_{max}$	$1.875 \times G_{max}$
Others	$1.25 \times G_{max}$	$1.25 \times G_{max}$

Appendix B

Examples of Some Outputs

```
......
11
00:00:40,951 --> 00:00:43,790
basically you should know it already how it works.
基本上你应该已经知道它，它是如何工作。

12
00:00:44,281 --> 00:00:45,570
we want to do this year,
我们想要做这一年，

13
00:00:45,611 --> 00:00:49,660
select city, and count from the world population
选择城市，再从世界人口数

14
00:00:49,661 --> 00:00:51,220
where gender is male.
那里的性别是男的。
......
```

Listing B.1: Automatically produced bilingual subtitle in .srt format with imperfect accuracy

```
......
——Supercomputing as a Service is when ... 00:07:49
$$ THINGS CHANGE OVER TIME 00:09:36
—$$ Trends and Accepted Customs in Supercomputing 00:10:16
——Speed doubles every 18 months 00:10:16
——There is no need to work on speed 00:10:16
——Next System is 10 − 100 times faster 00:10:16
——User will get 10 − 100 times faster solution 00:10:16
——Hardware develops faster than Software 00:10:16
——Software optimization is not worth doing HPC is an arms race 00:10:16
——No need to argue about funding 00:10:16
—$$ The Future of HPC 00:13:29
——Everything gets denser 00:13:29
———Everything gets hotter 00:13:29
———Everything gets heavier 00:13:29
———Everything gets more complex 00:13:29
———Renew your power supply and cooling! 00:13:29
——We move from 1 ton to several tons 00:13:29
......
```

Listing B.2: Outline outputed by TOG (*with "$$" as the sign of slide title*)

```
......
2 1 Supercomputing as a Service is when ...
0 0 THINGS CHANGE OVER TIME
1 0 Trends and Accepted Customs in Supercomputing
2 1 Speed doubles every 18 months
2 1 There is no need to work on speed
2 1 Next System is 10 − 100 times faster
2 1 User will get 10 − 100 times faster solution
2 1 Hardware develops faster than Software
2 1 Software optimization is not worth doing HPC is an arms race
2 1 No need to argue about funding
1 0 The Future of HPC
2 1 Everything gets denser
3 2 Everything gets hotter
3 2 Everything gets heavier
3 2 Everything gets more complex
3 2 Renew your power supply and cooling!
2 1 We move from 1 ton to several tons
......
```

Listing B.3: Outline reformatted for easier implementation (*two numbers are general hierarchy and intra-slide hierarchy accordingly*)

```
00:00:01 <casual> HPC and Clouds - How do we proceed?
00:01:28 <logical>   CLOUDS
00:06:07 <logical>   SUPERCOMPUTING
00:09:36 <logical>   THINGS CHANGE OVER TIME
00:20:00 <logical>   SCENARIOS
00:23:08 <ending> Questions Michael M. Resch
```

Listing B.4: Lecture video segmentation result for implementation with corresponding starting time and segment type. Example outline for Listing B.2 & B.3 & B.4 is extracted from lecture 6663 on tele-TASK.de – copyright belongs to original presenter: Prof. Dr. Michael M. Resch.

References

[1] Sherry, L., et al.: Issues in distance learning. International journal of educational telecommunications 1(4), 337–365 (1996) 1

[2] Simpson, O.: Supporting students in online open and distance learning. Routledge (2013) 1

[3] Schillings, V., Meinel, C.: tele-task: teleteaching anywhere solution kit. In: Proceedings of the 30th annual ACM SIGUCCS conference on User services. pp. 130–133. ACM (2002) 1

[4] Grünewald, F., Yang, H., Mazandarani, E., Bauer, M., Meinel, C.: Next generation tele-teaching: Latest recording technology, user engagement and automatic metadata retrieval. In: Human Factors in Computing and Informatics, pp. 391–408. Springer (2013) 1

[5] McAuley, A., Stewart, B., Siemens, G., Cormier, D.: The mooc model for digital practice (2010) 2

[6] Pappano, L.: The year of the mooc. The New York Times 2(12), 2012 (2012) 2

[7] Meinel, C., Totschnig, M., Willems, C.: openhpi: Evolution of a mooc platform from lms to soa. In: Proceedings of the 5th International Conference on Computer Supported Education (CSEDU), INSTICC, Aachen, Germany. vol. 5 (2013) 2

[8] Meinel, C., Willems, C.: openHPI: das MOOC-Angebot des Hasso-Plattner-Instituts, vol. 79. Universitätsverlag Potsdam (2013) 2

[9] Arbaugh, J.: How classroom environment and student engagement affect learning in internet-based mba courses. Business Communication Quarterly 63(4), 9–26 (2000) 3

[10] Ary, E.J., Brune, C.W.: A comparison of student learning outcomes in traditional and online personal finance courses. Journal of Online Learning and Teaching 7(4), 465–474 (2011) 3

[11] Guo, P.J., Kim, J., Rubin, R.: How video production affects student engagement: An empirical study of mooc videos. In: Proceedings of the first ACM conference on Learning@scale conference. pp. 41–50. ACM (2014) 3

[12] Schell, M.: How to globalize online course content. In: Globalized e-learning cultural challenges, pp. 155–167. IGI Global (2007) 11

[13] Colas, J.F., Sloep, P.B., Garreta-Domingo, M.: The effect of multilingual facilitation on active participation in moocs. The International Review of Research in Open and Distributed Learning 17(4) (2016) 11

[14] Ostashewski, N., Thorpe, M., Gibson, D.: Addressing the challenges of a bilingually delivered online course: design and development of the australia china trade (act) mooc. In: Proc. World Conference on e-Learning in Corporate. Government. Healthcare and Higher Education. pp. 1284–1289. No. 1 (2013) 11

[15] Beaven, T., Comas-Quinn, A., Hauck, M., de los Arcos, B., Lewis, T.: The open translation mooc: creating online communities to transcend linguistic barriers. Journal of Interactive Media in Education 2013(3) (2013) 11

[16] AlDahdouh, A.A., Osório, A.J.: Planning to design mooc? think first! The Online Journal of Distance Education and e-Learning 4(2), 47 (2016) 11

[17] Kurch, A., Mälzer, N., Münch, K.: Qualitätsstudie zu live-untertitelungen–am beispiel des "tv-duells? (2015) 11

[18] Mamgain, N., Sharma, A., Goyal, P.: Learner's perspective on video-viewing features offered by mooc providers: Coursera and edx. In: MOOC,

Innovation and Technology in Education (MITE), 2014 IEEE International Conference on. pp. 331–336. IEEE (2014) 11

[19] Wu, S., Fitzgerald, A., Witten, I.H.: Second language learning in the context of moocs. In: CSEDU 2014. vol. 1, pp. 354–359. SCITEPRESS (2014) 12

[20] Fichten, C.S., Ferraro, V., Asuncion, J.V., Chwojka, C., Barile, M., Nguyen, M.N., Klomp, R., Wolforth, J.: Disabilities and e-learning problems and solutions: An exploratory study. Educational Technology & Society 12(4), 241–256 (2009) 12

[21] Stevenson, M., Gaizauskas, R.: Experiments on sentence boundary detection. In: Proceedings of the sixth conference on Applied natural language processing. pp. 84–89. Association for Computational Linguistics (2000) 12

[22] Matusov, E., Hillard, D., Magimai-Doss, M., Hakkani-Tür, D.Z., Ostendorf, M., Ney, H.: Improving speech translation with automatic boundary prediction. In: INTERSPEECH. vol. 7, pp. 2449–2452 (2007) 12

[23] Fügen, C., Kolss, M.: The influence of utterance chunking on machine translation performance. In: INTERSPEECH. pp. 2837–2840 (2007) 12

[24] Kirkland, C.E.: Evaluation of captioning features to inform development of digital television captioning capabilities. American annals of the deaf 144(3), 250–260 (1999) 12

[25] Gottlieb, H.: Subtitles–readable dialogue? Eye Tracking in Audiovisual Translation pp. 37–81 12

[26] Pražák, A., Psutka, J.V., Hoidekr, J., Kanis, J., Müller, L., Psutka, J.: Automatic online subtitling of the czech parliament meetings. In: International Conference on Text, Speech and Dialogue. pp. 501–508. Springer (2006) 13

[27] Ortega, A., Laínez, J.E.G., Miguel, A., Lleida, E.: Real-time live broadcast news subtitling system for spanish. In: INTERSPEECH. pp. 2095–2098 (2009) 13

[28] Sridhar, R., Aravind, S., Muneerulhudhakalvathi, H., Senthur, M.S.: A hybrid approach for discourse segment detection in the automatic subtitle generation of computer science lecture videos. In: Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on. pp. 284–287. IEEE (2014) 13

[29] Álvarez, A., Arzelus, H., Etchegoyhen, T.: Towards customized automatic segmentation of subtitles. In: Advances in Speech and Language Technologies for Iberian Languages, pp. 229–238. Springer (2014) 13

[30] Álvarez Muniain, A., Balenciaga, M., Pozo Echezarreta, A.d., Arzelus Irazusta, H., Matamala, A., Martínez Hinarejos, C.D.: Impact of automatic segmentation on the quality, productivity and self-reported post-editing effort of intralingual subtitles. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). pp. 3049–3053 (2016) 13

[31] Eizmendi, G., et al.: Automatic speech recognition for live tv subtitling for hearing-impaired people. Challenges for Assistive Technology: AAATE 07 20, 286 (2007) 13

[32] Aliprandi, C., Scudellari, C., Gallucci, I., Piccinini, N., Raffaelli, M., del Pozo, A., Álvarez, A., Arzelus, H., Cassaca, R., Luis, T., et al.: Automatic live subtitling: state of the art, expectations and current trends. In: Proceedings of NAB Broadcast Engineering Conference: Papers on Advanced Media Technologies, Las Vegas (2014) 13

[33] Gravano, A., Jansche, M., Bacchiani, M.: Restoring punctuation and capitalization in transcribed speech. In: Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on. pp. 4741–4744. IEEE (2009) 13

[34] Lu, W., Ng, H.T.: Better punctuation prediction with dynamic conditional random fields. In: Proceedings of the 2010 conference on empirical methods in natural language processing. pp. 177–186. Association for Computational Linguistics (2010) 13

[35] Ueffing, N., Bisani, M., Vozila, P.: Improved models for automatic punctuation prediction for spoken and written text. In: INTERSPEECH. pp. 3097–3101 (2013) 13, 14, 33

[36] Zhang, D., Wu, S., Yang, N., Li, M.: Punctuation prediction with transition-based parsing. In: ACL (1). pp. 752–760 (2013) 13

[37] Xie, L., Xu, C., Wang, X.: Prosody-based sentence boundary detection in chinese broadcast news. In: Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on. pp. 261–265. IEEE (2012) 13

[38] Levy, T., Silber-Varod, V., Moyal, A.: The effect of pitch, intensity and pause duration in punctuation detection. In: Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of. pp. 1–4. IEEE (2012) 13, 16

[39] Sinclair, M., Bell, P., Birch, A., McInnes, F.: A semi-markov model for speech segmentation with an utterance-break prior. In: INTERSPEECH. pp. 2351–2355 (2014) 13, 14

[40] Zimmerman, M., Hakkani-Tür, D., Fung, J., Mirghafori, N., Gottlieb, L., Shriberg, E., Liu, Y.: The icsi+ multilingual sentence segmentation system. Tech. rep., DTIC Document (2006) 13, 14

[41] Kolár, J., Lamel, L.: Development and evaluation of automatic punctuation for french and english speech-to-text. In: INTERSPEECH. pp. 1376–1379 (2012) 14

[42] Hasan, M., Doddipatla, R., Hain, T.: Multi-pass sentence-end detection of lecture speech. In: INTERSPEECH. pp. 2902–2906 (2014) 14

[43] Lee, A., Glass, J.R.: Sentence detection using multiple annotations. In: INTERSPEECH. pp. 1848–1851 (2012) 14, 16

[44] Khomitsevich, O., Chistikov, P., Krivosheeva, T., Epimakhova, N., Chernykh, I.: Combining prosodic and lexical classifiers for two-pass punctuation detection in a russian asr system. In: Speech and Computer, pp. 161–169. Springer (2015) 14, 16

[45] Xu, C., Xie, L., Huang, G., Xiao, X., Chng, E., Li, H.: A deep neural network approach for sentence boundary detection in broadcast news. In: INTERSPEECH. pp. 2887–2891 (2014) 14

[46] Cho, E., Kilgour, K., Niehues, J., Waibel, A.: Combination of nn and crf models for joint detection of punctuation and disfluencies. In: Sixteenth Annual Conference of the International Speech Communication Association (2015) 14, 16, 17

[47] Ivarsson, J., Carroll, M.: Code of good subtitling practice. Language Today, April (1998) 14, 39

[48] Karamitroglou, F.: A proposed set of subtitling standards in europe. Translation journal 2(2), 1–15 (1998) 14, 39

[49] PIPERIDIS, S., DEMIROS, I., PROKOPIDIS, P.: Infrastructure for a multilingual subtitle generation system. Linguistics in the Twenty First Century p. 369 (2009) 14

[50] Wambacq, P., Demuynck, K.: Efficiency of speech alignment for semi-automated subtitling in dutch. In: International Conference on Text, Speech and Dialogue. pp. 123–130. Springer (2011) 14

[51] Daelemans, W., Höthker, A., Sang, E.F.T.K.: Automatic sentence simplification for subtitling in dutch and english. In: LREC (2004) 14

[52] Luotolahti, J., Ginter, F.: Sentence compression for automatic subtitling. In: Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania. pp. 135–143. No. 109, Linköping University Electronic Press (2015) 14

[53] Álvarez, A., del Pozo, A., Arruti, A.: Apyca: Towards the automatic subtitling of television content in spanish. In: Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on. pp. 567–574. IEEE (2010) 14

[54] Sawaf, H.: Automatic speech recognition and hybrid machine translation for high-quality closed-captioning and subtitling for video broadcast. Proceedings of Association for Machine Translation in the Americas–AMTA (2012) 14

[55] Valor Miró, J., Spencer, R.N., Pérez González de Martos, A., Garcés Díaz-Munío, G., Turró, C., Civera, J., Juan, A.: Evaluating intelligent interfaces for post-editing automatic transcriptions of online video lectures. Open Learning: The Journal of Open, Distance and e-Learning 29(1), 72–85 (2014) 14

[56] De Sousa, S.C., Aziz, W., Specia, L.: Assessing the post-editing effort for automatic and semi-automatic translations of dvd subtitles. In: RANLP. pp. 97–103 (2011) 14

[57] Shriberg, E., Stolcke, A., Hakkani-Tür, D., Tür, G.: Prosody-based automatic segmentation of speech into sentences and topics. Speech communication 32(1), 127–154 (2000) 14

[58] Favre, B., Hakkani-Tür, D., Petrov, S., Klein, D.: Efficient sentence segmentation using syntactic features. In: Spoken Language Technology Workshop, 2008. SLT 2008. IEEE. pp. 77–80. IEEE (2008) 14

[59] Batista, F., Moniz, H., Trancoso, I., Mamede, N.: Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts. Audio, Speech, and Language Processing, IEEE Transactions on 20(2), 474–485 (2012) 14

[60] Wang, X., Ng, H.T., Sim, K.C.: Dynamic conditional random fields for joint sentence boundary and punctuation prediction. In: INTERSPEECH. pp. 1384–1387 (2012) 14

[61] Liu, Y., Stolcke, A., Shriberg, E., Harper, M.: Using conditional random fields for sentence boundary detection in speech. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 451–458. Association for Computational Linguistics (2005) 14

[62] Tilk, O., Alumäe, T.: Lstm for punctuation restoration in speech transcripts. In: Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH) (2015) 14, 33, 34, 37

[63] Gotoh, Y., Renals, S.: Sentence boundary detection in broadcast speech transcripts (2000) 16, 37

[64] Liu, Y., Shriberg, E., Stolcke, A., Hillard, D., Ostendorf, M., Harper, M.: Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. Audio, Speech, and Language Processing, IEEE Transactions on 14(5), 1526–1540 (2006) 16

[65] Pappu, A., Stent, A.: Automatic formatted transcripts for videos. In: Sixteenth Annual Conference of the International Speech Communication Association (2015) 16

[66] Hinton, G.E.: Learning distributed representations of concepts. In: Proceedings of the eighth annual conference of the cognitive science society. vol. 1, p. 12. Amherst, MA (1986) 16

[67] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research 3(Feb), 1137–1155 (2003) 17, 26

[68] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research 12(Aug), 2493–2537 (2011) 17

[69] Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: HLT-NAACL. pp. 746–751 (2013) 17, 27

[70] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12, 1532–1543 (2014) 17, 27

[71] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015) 17

[72] Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014) 17, 20

[73] Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (ACL) (2014) 17

[74] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014) 17

[75] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998) 17

[76] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997) 17

[77] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1), 1929–1958 (2014) 20

[78] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 675–678. ACM (2014) 22

[79] Harris, Z.S.: Distributional structure. Word 10(2-3), 146–162 (1954) 27

[80] Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Volume 2. year = 2014, publisher = The Association for Computer Linguistics 27

[81] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016) 27

[82] Goldhahn, D., Eckart, T., Quasthoff, U.: Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In: LREC. pp. 759–765 (2012) 27

[83] Chang, P.C., Galley, M., Manning, C.D.: Optimizing chinese word segmentation for machine translation performance. In: Proceedings of the third workshop on statistical machine translation. pp. 224–232. Association for Computational Linguistics (2008) 31

[84] Zheng, X., Chen, H., Xu, T.: Deep learning for chinese word segmentation and pos tagging. In: EMNLP. pp. 647–657 (2013) 31

[85] Chen, X., Qiu, X., Zhu, C., Liu, P., Huang, X.: Long short-term memory neural networks for chinese word segmentation. In: EMNLP. pp. 1197–1206 (2015) 31

[86] Chen, X., Xu, L., Liu, Z., Sun, M., Luan, H.B.: Joint learning of character and word embeddings. In: IJCAI. pp. 1236–1242 (2015) 31

[87] Christensen, H., Gotoh, Y., Renals, S.: Punctuation annotation using statistical prosody models. In: ISCA Tutorial and Research Workshop (ITRW) on Prosody in Speech Recognition and Understanding (2001) 37

[88] Kolář, J., Švec, J., Psutka, J.: Automatic punctuation annotation in czech broadcast news speech. SPECOM´ 2004 (2004) 37

[89] Saon, G., Kuo, H.K.J., Rennie, S., Picheny, M.: The ibm 2015 english conversational telephone speech recognition system. In: Sixteenth Annual Conference of the International Speech Communication Association (2015) 38

[90] Garcia, D., Ball, M., Parikh, A.: L@ s 2014 demo: best practices for mooc video. In: Proceedings of the first ACM conference on Learning@ scale conference. pp. 217–218. ACM (2014) 39

[91] Krauth, W.: Coming home from a mooc. Computing in Science & Engineering 17(2), 91–95 (2015) 39

[92] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. In: Soviet physics doklady. vol. 10, p. 707 (1966) 42

[93] Bateman, S., Brooks, C., Mccalla, G., Brusilovsky, P.: Applying collaborative tagging to e-learning. In: Proceedings of the 16th international world wide web conference (WWW2007) (2007) 47

[94] Rahimi, E., van den Berg, J., Veen, W.: A roadmap for building web2.0-based personal learning environments in educational settings. In: Proceedings of the fourth international conference on Personal Learning Environments (The PLE Conference 2013) (2013) 47

[95] Grünewald, F., Meinel, C.: Implementation and evaluation of digital e-lecture annotation in learning groups to foster active learning. IEEE Transactions on Learning Technologies 8(3), 286–298 (2015) 47

[96] Sack, H., Waitelonis, J.: Automated annotations of synchronized multimedia presentations. In: In Proceedings of the ESWC 2006 Workshop on Mastering the Gap: From Information Extraction to Semantic Representation, CEUR Workshop Proceedings. Citeseer (2006) 47

[97] Imran, A.S., Rahadianti, L., Cheikh, F.A., Yayilgan, S.Y.: Semantic tags for lecture videos. In: Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on. pp. 117–120. IEEE (2012) 47

[98] Grabe, M., Christopherson, K.: Optional student use of online lecture resources: resource preferences, performance and lecture attendance. Journal of Computer Assisted Learning 24(1), 1–10 (2008) 47

[99] Lonn, S., Teasley, S.D.: Saving time or innovating practice: Investigating perceptions and uses of learning management systems. Computers & Education 53(3), 686–694 (2009) 47

[100] Levasseur, D.G., Kanan Sawyer, J.: Pedagogy meets powerpoint: A research review of the effects of computer-generated slides in the classroom. The Review of Communication 6(1-2), 101–123 (2006) 48

[101] Hill, A., Arford, T., Lubitow, A., Smollin, L.M.: "i'm ambivalent about it" the dilemmas of powerpoint. Teaching Sociology 40(3), 242–256 (2012) 48

[102] Yang, H., Meinel, C.: Content based lecture video retrieval using speech and video text information. IEEE Transactions on Learning Technologies 7(2), 142–154 (2014) 48

[103] Repp, S., Waitelonis, J., Sack, H., Meinel, C.: Segmentation and annotation of audiovisual recordings based on automated speech recognition. In: International Conference on Intelligent Data Engineering and Automated Learning. pp. 620–629. Springer (2007) 49

[104] Zhang, J., Chan, R.H.Y., Fung, P., Cao, L.: A comparative study on speech summarization of broadcast news and lecture speech. In: Interspeech. pp. 2781–2784 (2007) 49

[105] Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015) 49

[106] Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al.: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023 (2016) 49

[107] Onishi, M., Izumi, M., Fukunaga, K.: Blackboard segmentation using video image of lecture and its applications. In: Pattern Recognition, 2000. Proceedings. 15th International Conference on. vol. 4, pp. 615–618. IEEE (2000) 49

[108] Yeh, F.H., Lee, G.C., Chen, Y.J., Liao, C.H.: Robust handwriting extraction and lecture video summarization. In: Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2014 Tenth International Conference on. pp. 357–360. IEEE (2014) 49

[109] Atapattu, T., Falkner, K., Falkner, N.: Automated extraction of semantic concepts from semi-structured data: Supporting computer-based education through the analysis of lecture notes. In: International Conference on Database and Expert Systems Applications. pp. 161–175. Springer (2012) 49

[110] Li, H., Dong, A.: Hierarchical segmentation of presentation videos through visual and text analysis. In: Signal Processing and Information Technology, 2006 IEEE International Symposium on. pp. 314–319. IEEE (2006) 49

[111] Yang, H., Gruenewald, F., Meinel, C.: Automated extraction of lecture outlines from lecture videos-a hybrid solution for lecture video indexing. In: CSEDU (1). pp. 13–22 (2012) 49

[112] Liu, Y., Bai, K., Mitra, P., Giles, C.L.: Tableseer: automatic table metadata extraction and searching in digital libraries. In: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries. pp. 91–100. ACM (2007) 49

[113] Fang, J., Gao, L., Bai, K., Qiu, R., Tao, X., Tang, Z.: A table detection method for multipage pdf documents via visual seperators and tabular structures. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on. pp. 779–783. IEEE (2011) 49

[114] Gatos, B., Danatsas, D., Pratikakis, I., Perantonis, S.J.: Automatic table detection in document images. In: Pattern Recognition and Data Mining, pp. 609–618. Springer (2005) 49

[115] Kasar, T., Barlas, P., Adam, S., Chatelain, C., Paquet, T.: Learning to detect tables in scanned document images using line information. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. pp. 1185–1189. IEEE (2013) 49

[116] Wang, Y., Phillips, I.T., Haralick, R.: Automatic table ground truth generation and a background-analysis-based table structure extraction method. In: Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on. pp. 528–532. IEEE (2001) 49

[117] Mandal, S., Chowdhury, S., Das, A.K., Chanda, B.: A simple and effective table detection system from document images. International Journal of Document Analysis and Recognition (IJDAR) 8(2-3), 172–182 (2006) 49

[118] Göbel, M., Hassan, T., Oro, E., Orsi, G.: Icdar 2013 table competition. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. pp. 1449–1453. IEEE (2013) 49, 58, 59

[119] Kieninger, T.G.: Table structure recognition based on robust block segmentation. In: Photonics West'98 Electronic Imaging. pp. 22–32. International Society for Optics and Photonics (1998) 51

[120] Shin, J., Guerette, N.: Table recognition and evaluation. In: Class of 2005 Senior Conference on Natural Language Processing (2005) 51

[121] Koprinska, I., Carrato, S.: Temporal video segmentation: A survey. Signal processing: Image communication 16(5), 477–500 (2001) 51

[122] Del Fabro, M., Böszörmenyi, L.: State-of-the-art and future challenges in video scene detection: a survey. Multimedia systems 19(5), 427–454 (2013) 51

[123] Chou, H.P., Wang, J.M., Fuh, C.S., Lin, S.C., Chen, S.W.: Automated lecture recording system. In: System Science and Engineering (ICSSE), 2010 International Conference on. pp. 167–172. IEEE (2010) 51

[124] Ram, A.R., Chaudhuri, S.: Media for distance education. In: Video Analysis and Repackaging for Distance Education, pp. 1–9. Springer (2012) 51

[125] Yamamoto, N., Ogata, J., Ariki, Y.: Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition. In: European Conference on Speech Communication and Technology. pp. 961–964 (2003) 51

[126] Lin, M., Nunamaker Jr, J.F., Chau, M., Chen, H.: Segmentation of lecture videos based on text: a method combining multiple linguistic features. In: System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on. pp. 9–pp. IEEE (2004) 51

[127] Bhatt, C.A., Popescu-Belis, A., Habibi, M., Ingram, S., Masneri, S., McInnes, F., Pappas, N., Schreer, O.: Multi-factor segmentation for topic visualization and recommendation: the must-vis system. In: Proceedings of the 21st ACM international conference on Multimedia. pp. 365–368. ACM (2013) 51

[128] Shah, R.R., Yu, Y., Shaikh, A.D., Tang, S., Zimmermann, R.: Atlas: automatic temporal segmentation and annotation of lecture videos based on modelling transition time. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 209–212. ACM (2014) 51

[129] Hermann, C., Ottmann, T.: Electures-wiki—toward engaging students to actively work with lecture recordings. IEEE Transactions on Learning Technologies 4(4), 315–326 (2011) 51

[130] Jeong, H.J., Kim, T.E., Kim, M.H.: An accurate lecture video segmentation method by using sift and adaptive threshold. In: Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia. pp. 285–288. ACM (2012) 51

[131] Schroth, G., Cheung, N.M., Steinbach, E., Girod, B.: Synchronization of presentation slides and lecture videos using bit rate sequences. In: Image Processing (ICIP), 2011 18th IEEE International Conference on. pp. 925–928. IEEE (2011) 51

[132] Li, K., Wang, J., Wang, H., Dai, Q.: Structuring lecture videos by automatic projection screen localization and analysis. IEEE transactions on pattern analysis and machine intelligence 37(6), 1233–1246 (2015) 51

[133] Shafait, F., Smith, R.: Table detection in heterogeneous documents. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. pp. 65–72. ACM (2010) 57

[134] Blanke, T., Bryant, M., Hedges, M.: Ocropodium: open source ocr for small-scale historical archives. Journal of Information Science 38(1), 76–86 (2012) 59

[135] Chattopadhyay, T., Sinha, P., Biswas, P.: Performance of document image ocr systems for recognizing video texts on embedded platform. In: Computational Intelligence and Communication Networks (CICN), 2011 International Conference on. pp. 606–610. IEEE (2011) 59

[136] Bell, K.E., Limber, J.E.: Reading skill, textbook marking, and course performance. Literacy Research and Instruction 49(1), 56–67 (2009) 87

[137] Fowler, R.L., Barker, A.S.: Effectiveness of highlighting for retention of text material. Journal of Applied Psychology 59(3), 358 (1974) 87

[138] Vacca, R.T., Vacca, J.A.L., Mraz, M.E.: Content area reading: Literacy and learning across the curriculum (2005) 87

[139] Armbruster, B.B., Anderson, T.H.: On selecting "considerate" content area textbooks. Remedial and Special Education 9(1), 47–52 (1988) 87

[140] Huffman, J.R., Cruickshank, R.D., Jambhekar, S.N., Van Myers, J., Collins, R.L.: Electronic book having highlighting feature (Sep 2 1997), uS Patent 5,663,748 87

[141] Chi, E.H., Hong, L., Gumbrecht, M., Card, S.K.: Scenthighlights: highlighting conceptually-related sentences during reading. In: Proceedings of the 10th international conference on Intelligent user interfaces. pp. 272–274. ACM (2005) 87

[142] Marshall, C.C.: Annotation: from paper books to the digital library. In: Proceedings of the second ACM international conference on Digital libraries. pp. 131–140. ACM (1997) 87

[143] Tashman, C.S., Edwards, W.K.: Active reading and its discontents: the situations, problems and ideas of readers. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2927–2936. ACM (2011) 87

[144] Scott, P.: Teacher talk and meaning making in science classrooms: A vygotskian analysis and review. Studies in Science Education 32(1), 45–80 (1998) 88

[145] Gibbs, G., Coffey, M.: The impact of training of university teachers on their teaching skills, their approach to teaching and the approach to learning of their students. Active learning in higher education 5(1), 87–100 (2004) 88

[146] Kim, J., Guo, P.J., Seaton, D.T., Mitros, P., Gajos, K.Z., Miller, R.C.: Understanding in-video dropouts and interaction peaks inonline lecture videos. In: Proceedings of the first ACM conference on Learning@scale conference. pp. 31–40. ACM (2014) 88

[147] Pickering, L.: The role of tone choice in improving ita communication in the classroom. TESOL Quarterly pp. 233–255 (2001) 89

[148] Chen, F.R., Withgott, M.: The use of emphasis to automatically summarize a spoken discourse. In: Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on. vol. 1, pp. 229–232. IEEE (1992) 90

[149] Arons, B.: Pitch-based emphasis detection for segmenting speech recordings. In: ICSLP (1994) 90

[150] Silverman, K.E.A.: The structure and processing of fundamental frequency contours. Ph.D. thesis, University of Cambridge (1987) 90

[151] Hirschberg, J., Grosz, B.: Intonational features of local and global discourse structure. In: Proceedings of the workshop on Speech and Natural Language. pp. 441–446. Association for Computational Linguistics (1992) 90

[152] Kochanski, G., Grabe, E., Coleman, J., Rosner, B.: Loudness predicts prominence: Fundamental frequency lends little. The Journal of the Acoustical Society of America 118(2), 1038–1054 (2005) 90

[153] Silipo, R., Greenberg, S.: Automatic transcription of prosodic stress for spontaneous english discourse. In: Proc. of the XIVth International Congress of Phonetic Sciences (ICPhS). vol. 3, p. 2351 (1999) 90

[154] Tamburini, F.: Automatic prosodic prominence detection in speech using acoustic features: an unsupervised system. In: INTERSPEECH (2003) 90

[155] Christodoulides, G., Avanzi, M.: An evaluation of machine learning methods for prominence detection in french. In: INTERSPEECH. pp. 116–119 (2014) 90

[156] Heldner, M., Strangert, E., Deschamps, T.: A focus detector using overall intensity and high frequency emphasis. In: Proc. of ICPhS. vol. 99, pp. 1491–1494 (1999) 90

[157] Fernandez, R., Ramabhadran, B.: Automatic exploration of corpus-specific properties for expressive text-to-speech: A case study in emphasis. In: 6th ISCA Workshop on Speech Synthesis (2007) 90

[158] Brenier, J.M., Cer, D.M., Jurafsky, D.: The detection of emphatic words using acoustic and lexical features. In: INTERSPEECH. pp. 3297–3300 (2005) 90

[159] Kakouros, S., Pelemans, J., Verwimp, L., Wambacq, P., Räsänen, O.: Analyzing the contribution of top-down lexical and bottom-up acoustic cues in the detection of sentence prominence. Interspeech 2016 pp. 1074–1078 (2016) 90

[160] Kennedy, L.S., Ellis, D.P.: Pitch-based emphasis detection for characterization of meeting recordings. In: Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on. pp. 243–248. IEEE (2003) 90

[161] Wrede, B., Shriberg, E.: Spotting ”hot spots” in meetings: human judgments and prosodic cues. In: INTERSPEECH (2003) 90

[162] Qian, X., Liu, G., Wang, Z., Li, Z., Wang, H.: Highlight events detection in soccer video using hcrf. In: Proceedings of the Second International Conference on Internet Multimedia Computing and Service. pp. 171–174. ACM (2010) 90

[163] Bach, N.H., Shinoda, K., Furui, S.: Robust highlight extraction using multi-stream hidden markov models for baseball video. In: IEEE International Conference on Image Processing 2005. vol. 3, pp. III–173. IEEE (2005) 90

[164] Ariki, Y., Kumano, M., Tsukada, K.: Highlight scene extraction in real time from baseball live video. In: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval. pp. 209–214. ACM (2003) 90

[165] Hanjalic, A.: Generic approach to highlights extraction from a sport video. In: Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on. vol. 1, pp. I–1. IEEE (2003) 91

[166] Huang, Y.F., Chen, W.C.: Rushes video summarization by audio-filtering visual features. International Journal of Machine Learning and Computing 4(4), 359 (2014) 91

[167] Zheng, Y., Zhu, G., Jiang, S., Huang, Q., Gao, W.: Visual-aural attention modeling for talk show video highlight detection. In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 2213–2216. IEEE (2008) 91

[168] Wang, F., Merialdo, B.: Multi-document video summarization. In: 2009 IEEE International Conference on Multimedia and Expo. pp. 1326–1329. IEEE (2009) 91

[169] Lu, S., Wang, Z., Mei, T., Guan, G., Feng, D.D.: A bag-of-importance model with locality-constrained coding based feature learning for video summarization. IEEE Transactions on Multimedia 16(6), 1497–1509 (2014) 91

[170] He, L., Sanocki, E., Gupta, A., Grudin, J.: Auto-summarization of audio-video presentations. In: Proceedings of the seventh ACM international conference on Multimedia (Part 1). pp. 489–498. ACM (1999) 91

[171] Taskiran, C.M., Pizlo, Z., Amir, A., Ponceleon, D., Delp, E.J.: Automated video program summarization using speech transcripts. IEEE Transactions on Multimedia 8(4), 775–791 (2006) 91

[172] Qi, Y., Hunt, B.R.: Voiced-unvoiced-silence classifications of speech using hybrid features and a network classifier. IEEE Transactions on Speech and Audio Processing 1(2), 250–255 (1993) 92

[173] Atal, B., Rabiner, L.: A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 24(3), 201–212 (1976) 92

[174] Deng, H., O'Shaughnessy, D.: Voiced-unvoiced-silence speech sound classification based on unsupervised learning. In: 2007 IEEE International Conference on Multimedia and Expo. pp. 176–179. IEEE (2007) 92

[175] Bachu, R., Kopparthi, S., Adapa, B., Barkana, B.D.: Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy. In: Advanced Techniques in Computing Sciences and Software Engineering, pp. 279–282. Springer (2010) 92

[176] Stoltzman, W.T.: Toward a social signaling framework: Activity and emphasis in speech. Ph.D. thesis, Massachusetts Institute of Technology (2006) 95

[177] Pousman, Z., Stasko, J.: A taxonomy of ambient information systems: four patterns of design. In: Proceedings of the working conference on Advanced visual interfaces. pp. 67–74. ACM (2006) 96

[178] Pan, M.H., Yamashita, N., Wang, H.C.: Task rebalancing: Improving multilingual communication with native speakers-generated highlights on automated transcripts. In: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing. pp. 310–321. ACM (2017) 97

[179] Luzanova, O.: Means of speech for the creation of a positive image of man in a panegyric discourse. typical deviations in english speech, made by non-native speakers (considering english personal advertisements) (2014) 98

[180] Crines, A., Heppell, T.: Rhetorical style and issue emphasis within the conference speeches of ukip's nigel farage 2010–2014. British Politics (2016) 98

[181] Pedrosa-de Jesus, H., da Silva Lopes, B.: Exploring the relationship between teaching and learning conceptions and questioning practices, towards academic development. Higher Education Research Network Journal p. 37 (2012) 99

[182] Li, M., Jiang, X.: Art appreciation instruction and changes of classroom questioning at senior secondary school in visual culture context. Cross-Cultural Communication 11(1), 43 (2015) 99

[183] Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS quarterly pp. 319–340 (1989) 101

[184] Natke, U., Grosser, J., Kalveram, K.T.: Fluency, fundamental frequency, and speech rate under frequency-shifted auditory feedback in stuttering and nonstuttering persons. Journal of Fluency Disorders 26(3), 227–241 (2001) 106

[185] Quené, H.: Multilevel modeling of between-speaker and within-speaker variation in spontaneous speech tempo. The Journal of the Acoustical Society of America 123(2), 1104–1113 (2008) 106

[186] O'Shaughnessy, D.: Timing patterns in fluent and disfluent spontaneous speech. In: Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on. vol. 1, pp. 600–603. IEEE (1995) 106

Acronyms

AF	Adaptive Features, used in TOG
ARD	Arbeitsgemeinschaft der öffentlich-rechtlichen Rundfunkanstalten der Bundesrepublik Deutschland
ASR	Automated Speech Recognition
CC	Closed Captioning
ConvNet	Convolutional Neural Network
CRF	Conditional Random Fields
DHCP	Dynamic Host Configuration Protocol
DL	Deep Learning
DNN	Deep Neural Network
DT	Decision Tree
GHG	General Hierarchical Gap, used in TOG
GLS	Globally Logical Segment, used in lecture video segmentation
HMM	Hidden Markov Model
HPI	Hasso Plattner Institute
HT-SU	Half-Textual Slide Unit, used in lecture highlighting
IB	Item Bullet, used in TOG
IPv6	Internet Protocol version 6
IT	Information Technology
IWSLT	International Workshop on Spoken Language Translation
LCS	Low Case Start, used in TOG
LDR	Levenshtein Distance Rate, used in TOG
LDR-T	Levenshtein Distance Rate of Titles, used in TOG
LM scores	Language Model scores
LMC	Lexical Model Configuration, used in SBD
LS	Logical Segment, used in lecture video segmentation
LSTM	Long-Short Term Memory
MOOC	Massive Open Online Course
MT	Machine Translation
n-TS	Negative Time Segment, used in lecture video segmentation
NDP	Neighbor Discovery Protocol
NLP	Natural Language Processing
NT-SU	Non-Textual Slide Unit, used in lecture highlighting
OCR	Optical Character Recognition
p-TS	Positive Time Segment, used in lecture video segmentation
PDF	Portable Document Format
PLS	Partially Logical Segment, used in lecture video segmentation
POS tags	Part-Of-Speech tags
PPTX	The default presentation file format for PowerPoint 2007 and newer
PTA	Potential Title Area, used in TOG
RNN	Recurrent Neural Network
SBD	Sentence Boundary Detection
STD	Slide Transition Detection
SU	Slide Unit, used in lecture highlighting
SVM	Support Vector Machine

SWR	Shared Word Rate, used in TOG
T-SU	Textual Slide Unit, used in lecture highlighting
TED	Technology, Entertainment, Design. http://www.ted.com/
tele-TASK	tele-Teaching Anywhere Solution Kit
TOG	Tree-structure Outline Generation
TS	Time Segment, used in lecture video segmentation
TV	Television
V/U/S	Voiced sound, Unvoiced sound and Silence
WER	Word Error Rate
WV	Word Vector
ZCR	Zero-Crossing Rate
ZDF	Das Zweite Deutsche Fernsehen

Acknowledgements

To everything comes an end.

I still remember my first day in HPI, when *Mrs. Michaela Schmitz* led me from the entrance of building to the office and then *Mr. Matthias Bauer* showed me around the chair to meet my new colleagues. Gradually, I became the "new" colleague to be met for the newer-comers and finally I could call it a day.

In this journey of pursuing Ph.D., the first person I need to thank is definitely my advisor, *Prof. Dr. Christoph Meinel*. He is not only a successful and reputable researcher, but also a motivated and fearless pioneer. What his broad horizon, rich experience and keen intuition have brought to me is the accurate general direction of my research, but more importantly, it is his enthusiasm and passion for work which always spurs on me to overcome the difficulty and keep moving. Thank you, professor, Danke Schön.

Then I would like to express my sincere gratitude to *Dr. Haojin Yang*, who's the senior researcher in the chair and the direct tutor to me. His suggestions in detail helped me so much not only in specific research topics, but also about how to organize the Ph.D. study. Along with *Cheng Wang* and *Sheng Luo*, this Chinese-speaking group gave me quite a lot of technical ideas and mental support. Thank you, buddies, 谢谢。

Next, my international colleagues, although most of them can still not pronounce my name acoustically correct. However, maybe my pronunciations of their names are also not correct. This tiny problem would not hinder our open-minded and friendly discussion about almost everything. I'm afraid I cannot list every name here, but these interesting people at least include 2 Matthias, 3 Martin, 4 Christian and so many others from Germany, Syria, Russia, Nigeria, Indonesia, Iran, Palestine, *etc.* Thank you all.

Besides, I also want to thank *Dr. Sharon Nemeth* for helping me with my English speaking and writing skills and the 4 German teachers who attempted to improve my German. And also my gratitude goes to our chair secretary Michaela, HPI receptionist Daniela and all the other HPI staff who have helped me. And the students, who were involved in the seminars I worked as the tutor, thank you either.

Another special person I need to thank is *Prof. Baocai Yin*, my advisor when pursuing master degree. Thank you for providing me such a great opportunity to study in HPI under Prof. Meinel. I also thank *Dr. Feng Cheng* for assisting in this process.

In the end, I must thank my family, who were about 9000 kilometers away from me. But as introduced in this thesis, geographical distance is no longer a barrier in this era of digitalization and globalization. I know exactly that for every single day, hour, minute or second, my family were, are and will be always with me. They are my most solid backing.

Well, as I said, it is time to call it a day. Thank you, HPI, thank you, Potsdam, thank you, Germany. Ph.D. is only a step in the journey of life and I will keep moving.

To everything comes a beginning.