# MASTERARBEIT / MASTER’S THESIS

Titel der Masterarbeit / Title of the Master‘s Thesis

## “Designing a CAT tool for subtitling environments”

verfasst von / submitted by

Alina Malaeva, BA

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Arts (MA)

Wien, 2017 / Vienna 2017

| | |
|---|---|
| Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet: | A 060 342 360 |
| Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet: | Masterstudium Übersetzen UG2002<br>English<br>Russisch |
| Betreut von / Supervisor: | Univ.-Prof. Mag. Dr. Gerhard Budin |

## Abstract

Over the past few years, subtitles have gained greater visibility with the public due to increasing digitalisation and internet use. Consequently, they are no longer confined to traditional spheres like films and TV and are increasingly a part of people's everyday experience. Yet, their variable quality and inconsistencies between translations of the same source material, not only in films and series, but also in e-learning, scientific, and marketing videos, demand for an investigation of the problem and its resolving.

The above observation is the starting point for the hypothesis presented in this thesis, namely, that there is a need for computer-aided translation (CAT) tools in subtitling environments. This thesis, therefore, designs a solution aimed at tackling translation problems in both intralingual and interlingual subtitling processes. The proposed solution aims to facilitate work for translators in the subtitling industry by simultaneously raising their productivity and output quality.

Section 1 presents essential information about current translation technology and audiovisual translation (AVT) as context for the hypothesis. That hypothesis is substantiated in Section 2 by arguments for such a tool from an academic and professional point of view, including some field tests of currently available tools. Section 3 identifies the software features required for my proposed solution and also examines useful solutions already on the market by analysing subtitle attributes and their translational needs. Section 4 experiments with the tools identified in Section 3 to suggest functionalities that a subtitling software should include in order to boost consistency, and thus quality, while at the same time increasing productivity for translators working in subtitling environments. My final conclusions are briefly presented in Section 5.

# Contents

## List of figures

# List of tables

## List of abbreviations

| Abbreviation | Full |
|---|---|
| **ASR** | automated speech recognition |
| **AVT** | audiovisual translation |
| **AVTEnT** | Audiovisual Translation Environment Tool |
| **CAT** | computer-aided translation, computer-assisted translation |
| **cps** | characters per second |
| **MAT** | machine-assisted translation |
| **MT** | machine translation |
| **OCR** | optical character recognition |
| **QA** | quality assurance |
| **RTF** | rich text format |
| **SDH** | subtitles for the deaf and hard-of-hearing |
| **SL** | source language |
| **SMT** | statistical machine translation |
| **TL** | target language |
| **TM** | translation memory |
| **TMS** | terminology management system |
| **UI** | user interface |
| **WER** | word error rate |
| **wpm** | words per minute |
| **XLIFF** | XML Localization Interchange File Format |

# 0. Introduction

A great number of technology is used in professional translation, such as web dictionaries, text-editing programs, translation memories (TMs) and many other tools and linguistic assets. The same is true also of subtitling and dubbing, where different software programs facilitate work. Yet, there is no translation memory that can be used within subtitling software. A host of other tools are used: automatic speech and dictation recognition software; automatic song recognition mobile applications; online solutions like Google Translate for both written and spoken language; CAT (computer-assisted translation) tools; machine translation; and subtitling software that recognises shot changes. How is it possible that there is still no CAT tool in the professional area of subtitling? And how do I know that there is none? Here is a telling example that highlights the lack of such a tool: ‘Eye on Springfield’ is a fictional current affairs show and recurs in the TV series *The Simpsons*. It appears several times in many different episodes and is always called ‘Eye on Springfield’, at least in the original US series. For the German version, it was translated from scratch for each episode. The outcome is interesting: ‘Alle Augen auf Springfield’ (DVD subtitle: ‘Springfield im Visier’), ‘Augenzeuge in Springfield’ (DVD subtitle: ‘Spot auf Springfield’), ‘Brennpunkt Springfield’ (DVD subtitle: ‘Brennpunkt Springfield’), etc. This obviously raises the question: How is it that the titles are always different? Working freelance in subtitling myself and being passionate about *The Simpsons*, I realised that there are elements in series that reappear in different parts of one or more episodes or in later seasons which have been translated anew, but in a different way. Since such elements as in-series show names or slogans have been translated in many ways, I started wondering what the reason for this could be.

Some preliminary research revealed that the problem must lie within the very process of translation. This process has been revolutionised since the 1980s by technology that not only facilitates and speeds up translation, but also reduces time and therefore costs thanks to CAT tools and TMs. So why is this technology not being used for TV purposes? One answer could be that the dialogue/dubbing director decided that the translation does not synchronise with the speaker’s lip movements and therefore needs to be reworded. But what about subtitles? Should they not always be the same? However, since subtitles are bound to time and space constraints for readability reasons, the translated subtitle can sometimes even differ from what is said in the dubbed version. Most production companies order translations from different suppliers depending on what is being translated: one company translates the audio and another company creates the subtitles, while yet another company is assigned to make closed captions for people with hearing impairments. To avoid such huge discrepancies, I would like to suggest the use of a translation memory and the addition of some extra functionalities to a CAT tool. This thesis will outline how that could be achieved.

But first it will be necessary to find out how often such a feature is needed and how usable it is within existing subtitling software. As for the methodology, it will consist of analysing the market situation, comparing existing CAT tools and subtitling programs, and finally, based on the results of that investigation, thinking of possible solutions for a translation memory in subtitling. At the same time, I will personally try out CAT solutions in subtitling environments that are already in place, if there are any, and document my findings. Accordingly, I have conducted some desk research on what products there are on the market and which functions they feature. Are there any ground-breaking developments? If not, what might the reason for this be? A person who works in a company providing subtitling software to whom I posed this question said that there is not enough money for R&D. Some other professionals in the subtitling industry I met were also not convinced that a TM can be useful, but the people I spoke to were in general not fond of using CAT tools. So, the issue may be the general low level of enthusiasm for technology among translators. I will not be able to provide answers to this question within this one thesis, but I hope to be able to offer a solid basis for further investigation. I will discuss possible solution options in Chapter 4 and elaborate a theoretical framework for a subtitling program with a CAT component, which can later be evaluated in further studies. I would like to take inspiration from Bowker and Fisher's (2010: 60) TEnt for Translation Environment Tool, and call the proposed tool AVTEnT (/ei vi tent/ like *evident* as in *it is evident that we need a CAT tool in subtitling*), short for Audiovisual Translation Environment Tool. Even though, as you will notice later in this thesis, there is much more to AVT than mere subtitling, I will elaborate only the voice-to-text feature of it.

Furthermore, I should point out that a) while, as a subtitling practitioner I can suggest the features of a fully functioning application from an end-user point of view, not being an engineer I cannot, of course, specify the precise construction design, and b) the focus of this thesis is on helping subtitlers and translators in their daily life. It is not about how to make professional subtitlers and/or translators obsolete in the industry, or what the easiest way is to lower prices per subtitle by giving discounts on fuzzy matches. The objective underpinning this paper's desk research, field testing and documentation is to propose a design solution to raise quality and consistency throughout subtitlers' work by also facilitating it, with no intention to praise or criticise any of the tools named herein.

# 1. CAT tools, AVT, and technology

This introductory section provides an overview of terms that are frequently used in this thesis to avoid misunderstandings or explanations within the text. Since this thesis is directed at a specialist audience, I will keep the definitions as concise as possible with further reading included in the sections to come.

## 1.1. Computer-aided translation tool (CAT tool)

Computer-aided translation tools, also known as computer-assisted translation tools (CAT tools), support translators in their translation processes. These tools can either refer to components that form a software package or software applications that can be used individually. CAT tools have their roots in the invention of machine translation (MT). Chan (2015: 3-31) offers a concise overview of the history of CAT tools and MT. In 1947, one year after the computer was invented, the scholars Locke and Boothe (1955) began to write memoranda that they had compiled: this became the first book on MT entitled Machine Translation of Languages: Fourteen Essays. In 1954, for the first time, Russian sentences were translated on the IBM701 machine, and the Massachusetts Institute of Technology (MIT) published the first journal on MT. The USA, however, did not reach the achievements they had expected, so the government set up the Automatic Language Processing Advisory Committee (ALPAC) to evaluate the situation for MT. The government came to the conclusion that MT would not deliver high-quality translations for a long time; nonetheless, ‘machine-aided translation may be an important avenue toward better, quicker, and cheaper translation’ (ALPAC 1966: 32). This was the first step towards CAT. In 1978, ALPS (Automated Language Processing Systems) was developed by Alan Melby, which was the first translation system that was designed to find full matches of previously translated segments: much like translation memories do nowadays. Over the years, several other attributes that an interactive translation system should have were also proposed (as cited in Chan 2015: 4); for example, ‘translation by text-retrieval’ by Arthern in 1979 (1979: 93) and a divided monitor for source and target texts as well as dictionary consultation by Kay in 1980. Additionally, in the same year, researchers tried to collect translation samples to build a bilingual database but, due to technology’s limited development at the time, no CAT system was commercially available. Indeed, it was not until 1984 that the first CAT companies—Trados and STAR Group (Software Translation Artwork Recording AG)—were founded (Chan 2015: 4-6). Both companies still operate in the market and have influenced the idea for this thesis through their development of the translation programs SDL Trados Studio and STAR's Transit NXT.

The main elements that comprise a CAT tool include a translation memory system, a term base, and a terminology management system with active terminology recognition, pre-translation features (like a machine translation system), a concordance search function, a project management module, a document analysis module that counts words, segments and fuzzy matches, an alignment module to link source text sections with their translations, and finally, a quality control module. All these features facilitate the translation process and are vital in the daily business of a translator. Since these features are all useful, but not relevant to the subject of this thesis, the following sections discuss only those features that are pertinent to subtitling environments.

### 1.1.1. Translation memory (TM)

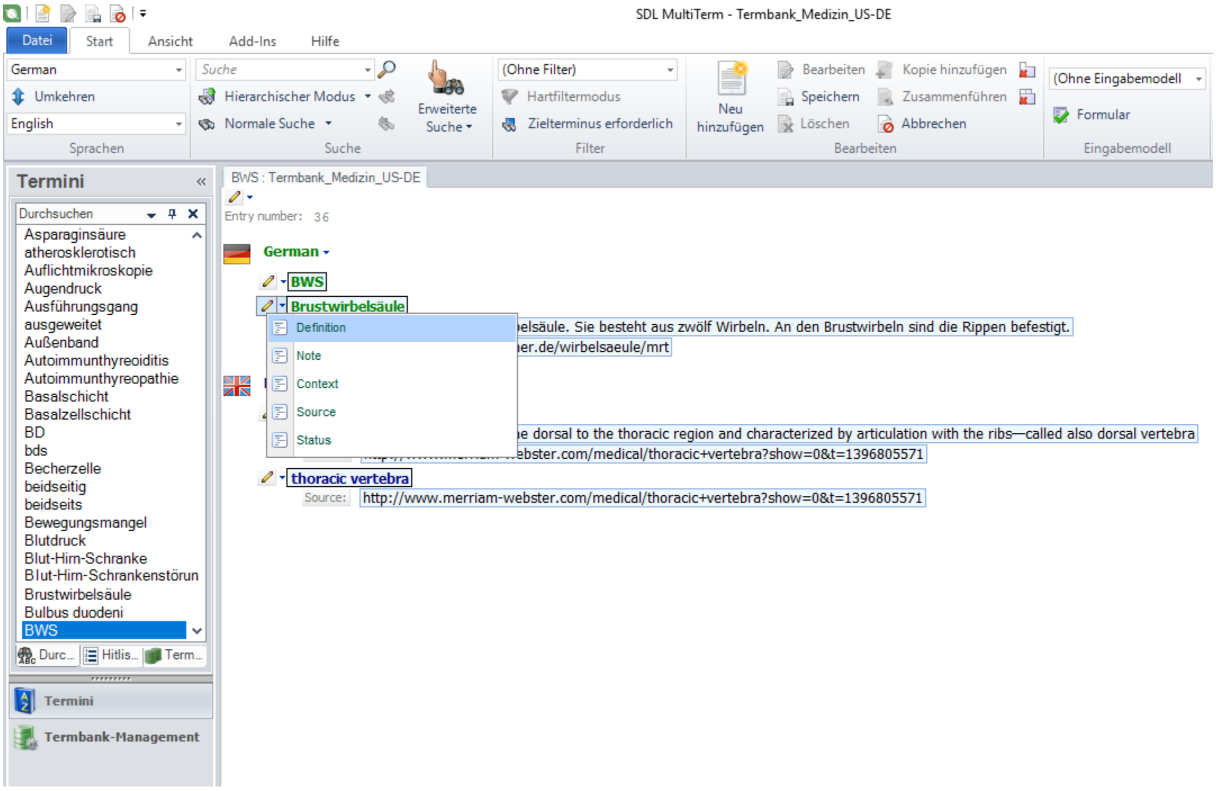
A translation memory (TM) is one of the basic functionalities of every CAT tool. This is a database with the written source segment stored together with its translation in a translation unit, which is defined by the TMX Standard (LISA 2001) as 'an entry consisting of aligned segments of text in two or more languages.' Thus, the translator can retrieve the paired match for reuse on repetitions or similar segments (so-called fuzzy matches) within the source text. CAT tools thereby can help to increase a translator's productivity and ensure consistency throughout the text. For translation memories, one premise is that the text to be translated is available in (or can be converted into) written and electronic form; for example, by means of optical character recognition (OCR) software. The OCR software recognises character patterns in a scanned image, compares them to patterns stored in its database and gives an output of the best match in text format. This technology is very useful for translators as this helps them to quickly process a hard copy or PDF and feed it into their CAT tool, which saves time and costs compared to typing the text in (see for more on this in *4.1.1 OCR*).

There are many CAT tool providers. The choice of products ranges from online to offline and from pay-per-use to proprietary solutions: but the core function is the TM. This TM can be extracted and sent to (for instance) the customer or a colleague for reference. However, since there are many providers, there are also many different file formats for TMs, such as the native **.sdltm* extension in the SDL Trados Studio software. To enable an exchange of TM files between users of different software applications, standard exchange formats have been set up, such as the XLIFF (XML Localization Interchange File Format) standard by the Organization for the Advancement of Structured Information Standards

(OASIS[1]) and the above mentioned TMX (Translation Memory eXchange) file format by OSCAR, the special interest group of the now insolvent LISA. Why I consider a TM useful for subtitling and the challenges facing a tool like this in subtitling is discussed in Section 3.3.

### 1.1.2. Term base and terminology management system (TMS)

A Terminology Management System (TMS) is used to store terminology in a term base. Basically, term bases serve as multi-lingual glossaries where translators can retrieve corresponding terminological translations. A term base is a database that can be built up by a translator or downloaded from the internet on various subjects and in various languages.[2] It usually contains vocabulary of specialised domains, but can also be used as an electronic dictionary (see Bowker 2002: 81, 2002: 101). The translator can create a term base for a certain domain and fill it with entries by typing one term in one language, filling in all relevant attributes including the domain label, linking it to its translations in different languages and storing it, as shown in *Figure 1*.



Entry number: 36

German

BWS

Brustwirbelsäule

elsäule. Sie besteht aus zwölf Wirbeln. An den Brustwirbeln sind die Rippen befestigt.

er.de/wirbelsaeule/mrt

e dorsal to the thoracic region and characterized by articulation with the ribs—called also dorsal vertebra

ebster.com/medical/thoracic+vertebra?show=0&t=1396805571

thoracic vertebra

Source: http://www.merriam-webster.com/medical/thoracic+vertebra?show=0&t=1396805571

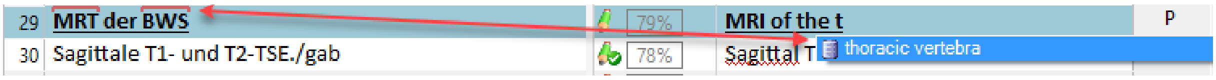
*Figure 1: Example of a term base entry in editing mode in SDL MultiTerm Desktop 2014 UI*

---

[1] See www.oasis-open.org (last accessed on 2017-06-06)

[2] For more see appstore.sdl.com (last accessed 2017-06-28)

The term can then be retrieved during the translation process within the translation editor if the term base has been linked to the project and terminology recognition is activated. For recognition, the term database scans through the source text and compares terms or groups of words in the source text against stored items in the term base. When there is a match, the terminology is then marked by the CAT tool in the SL column. Once the translator types the first letter of the translation in the TL field, the term base auto-suggests the fully translated term, as shown in *Figure 2*.



| | | | | |
|---|---|---|---|---|
| 29 | MRT der BWS | 79% | MRI of the t | P |
| 30 | Sagittale T1- und T2-TSE./gab | 78% | Sagittal T | |

*Figure 2: Screenshot of a terminology recognition in SDL Trados Studio 2014*

Since term bases are a useful component of the CAT tool, it is likewise essential in an AVT environment. I therefore elaborate on the function of term bases in *4 Solution: Which CAT features should the AVTEnT have?*

### 1.1.3. Alignment

An alignment tool is used to manually or automatically relate segments of a SL text to segments of a TL text of a document. The document can be uploaded to the module for segmentation and analysis. Text here can refer to a word, phrase, or sentence; hence, there is word alignment, phrase alignment, and sentence alignment. The alignment process can be completed automatically and double-checked manually by the translator. Usually, a manual correction is needed to make sure that the relation has been performed correctly. By means of a manual correction, the translator or alignment specialist raises quality levels for future translations. Quality is one of the issues this thesis addresses; consequently, the following component is equally important.

### 1.1.4. Quality management and quality assurance (QA)

Translators need to meet constantly tighter deadlines and often have to offer discounts on fuzzy and 100% matches while maintaining a high quality for their products and for less money. Pressure is high, competition is easily accessible on the web, and the bills must be paid whether the translator gets a job or not. If the translator’s output quality is low, he or she may face problems with customer satisfaction. This is an issue for quality management. To assist the translator with quality assurance, there are quality assurance component that are either built into CAT tool suites, like SDL Trados Studio, or can be downloaded as an add-on like ApSIC Xbench for a yearly subscription fee. While a built-in QA module is already helpful with identifying empty translation segments, identifying between SL and TL segments, inconsistency in punctuation, using tag elements between SL and TL segments, and spelling and grammar mistakes, an add-on like Xbench can be very powerful in big projects and crowdsourced translations. It can scan through many documents within many projects at once to spot an inconsistent translation or more than one translation for the same SL segment, and points out deviations from mandatory use of certain terms from the term base.

## 1.2. Audiovisual translation (AVT)

Audiovisual translation (AVT) has become more important in these last decades of growing mainstream media and internet usage. The main AVT modes are subtitling, dubbing, voiceover, and sign language interpreting, with audio description being the most recent mode. In this thesis, I focus on the mode of subtitling but provide a rough overview of various other modes below.

### 1.2.1. Dubbing

For dubbing, the following definition is clear and outlines the difficulties translators face during script translation for a film:

> "It involves replacing the original soundtrack containing the actor's dialogue with a target language (TL) recording that reproduces the original message, ensuring that the TL sounds and the actors' lip movements are synchronised in such a way that target viewers are led to believe that the actors on screen are actually speaking their language."
>
> (Díaz Cintas & Orero 2010: 442)

Lip synchrony is the distinguishing factor when compared to other AVT modes. Instead of aiming for a 100% synchrony however, a matching of 'bi-labial consonants /b/, /p/, /m/, the labio-dentals /f/, /v/, and some open vowels' (Fodor 1976 cited in Schwarz 2010: 399) is sufficient to satisfy the viewer's perception of synchrony.

As for translation processes, the preparation of a script for voice actors is the responsibility of dubbing directors, translators, and dialogue writers, which makes the whole process costly and requires good teamwork to meet the tight deadlines. The dubbing director then coordinates dubbing speakers during the recording of the translated dialogue in a special dubbing studio and edits it if the lip-movements do not synchronize with the soundtrack. This is time-consuming and costly; however, regardless of the costs, dubbing has been traditionally adopted as the preferred AVT mode in many European countries like Spain, Austria, Hungary, Germany, France, and Italy and in Asian countries like Japan and China (Chaume 2013: 107, Luyken 1991: 30). As Hillman has noted however, the internationalisation of films has also contributed to the growth of AVT processes and the above-mentioned teamwork deliver sometimes awkward results:

> "German films made in English, dubbed into German for local audiences, requiring retranslation back to English once shown abroad […]. We then finish up with subtitles synched with the lip movements of the speaker but not with the sounds we actually hear […]."
>
> (Hillman 2011: 386-7)

In addition to those countries with a tradition of dubbing, countries that have traditionally been subtitling such as Portugal, Denmark, and Norway and also Russia and Poland (despite its strong tradition in voiceover translation) are also currently shifting towards dubbing with foreign productions (Chaume 2013: 108).

Technology used in the field of dubbing extends from complimentary applications like Windows Movie Maker (also used for fandubs created by non-professional fans for web audiences) and freeware like Magix, Reaper (for PC and Mac), or Audacity (for Android) to licenced software such as Adobe Audition CC, which is included in Adobe's Creative Cloud for a monthly fee and features automatic speech alignment with automated dialogue replacement (ADR).

A possibly ground-breaking development in the field of dubbing is the program by Disney Research, which maps visemes with possible phonemes. This means that lip movements in a video can be synchronised with a range of alternative viseme matching word sequences (Taylor et al. 2015); the findings were presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2015. One example serves to illustrate the innovation of this program: when the speaker says clearly 'clean swatches', the software produces a list of 9658 alternative utterances. A synthesised voice then presents the possible alternatives and redubs the video with phrases such as 'like to watch you', 'need no pots' and many other utterances that match the lip movements.[3]

### 1.2.2. Voiceover (VO)

'From a translational perspective, voiceover consists in presenting orally a translation in a TL, which can be heard simultaneously over the SL voice.' (Díaz Cintas & Orero 2010: 441). This

---

[3] For more, please watch the video provided on https://www.disneyresearch.com/publication/a-mouth-full-of-words-visually-consistent-acoustic-redubbing/ (last accessed 2017-06-17)

means that for voiceover (VO) broadcasts, the SL soundtrack can be heard at reduced volume with the TL soundtrack superimposed at a higher volume so that both auditory channels can be heard at the same time. This AVT mode is used in documentaries where the translated audio starts with a slight delay after the original soundtrack; the possible intention here is to immerse the viewers in the film and to have them hear what is said in the original for credibility purposes (Chaume 2013: 108). For translation for VO, the translator usually works with the video and the script: the video's soundtrack is in language A, the script is a relay translation into language B and, for target translation, language C is needed (Díaz Cintas & Orero 2010: 442). As for costs, VO is cheaper in terms of production than dubbing (ibid.) and 'almost as cheap as subtitling' (Schwarz 2011: 402). In East European countries, such as Russia, Poland, Latvia, Belarus (among others) not only documentaries but also fictional films are traditionally translated in the VO mode. This is assumed to be due to the overly high costs for dubbing and low literacy rates for subtitling. In Russia during the Soviet period, the term *Gabrilov translation* was used for this AVT mode of VO and only one reader was used for all dialogues. Nowadays, there is one male reader for the main male role and one female reader for the main female role with perhaps a third person to read all other roles. One feature retained from the *Gabrilov translation* method is that dialogue is read in a monotonous way with no acting performed (Chaume 2013: 107).

### 1.2.3. Subtitling

Subtitling is the most commonly used AVT mode for audiovisual programmes because of its advantages of low costs and fast production. Subtitling is defined as 'rendering in writing the translation into a TL of the original dialogue exchanges uttered by the different speakers as well as of all other verbal information that is transmitted visually (letters, banners, inserts) or aurally (lyrics, voices off)' (Díaz Cintas 2010: 344). In this thesis however, the main differentiator is the target audience and for the purpose of this paper, TL in the above definition can be equal to the SL (*intralingual*), such as for SDH (subtitles for the deaf and hard-of-hearing people), or a different language (*interlingual*).

Since subtitles are used in various settings, such as for language learning, in educational videos, for film fandom (*fansubbing*), for the d/Deaf[4] and hard-of-hearing people (for people

---

[4] Deaf with a capital D is referred to people who identify themselves as members of a culture, community, and minority of Deaf and whose first language is not a spoken, but a signed language (Leeson & Vermeerbergen 2010, Pöchhacker 2004). Therefore, a spoken and written language may be a foreign language to them, which has to be considered in subtitling environments.

with both congenital and acquired hearing impairments), in film productions for film festivals, and also simply for internationalisation reasons, the translator faces different challenges in different target-audience related settings. Although SDH have to cover utterances on and off screen, include intonation, pauses, stuttering, music with artist and song title (as well as their lyrics in writing), and noises contributing to the understanding of the plot, not all of these aspects are applicable to subtitles for hearing audiences.

Professionally made subtitles are usually constrained to a certain length, a certain number of characters per line, number of lines, words per minute, all which are matched to the sounds and the visually transmitted images of the film. This means that a subtitle should not exceed a scene cut and should be synchronised with the dialogue without covering burnt-in subtitles like titles, place and date of a scene, inserts or pop-ups (see example in Figure 3). More on guidelines and standard values can be found in *2.1 Quality – guidelines and style* guides. Subtitles usually appear horizontally at the bottom or on the side of the screen for Japanese, but can also be displayed randomly as so-called *inserts* or 'integrated titles' (Fox 2012).

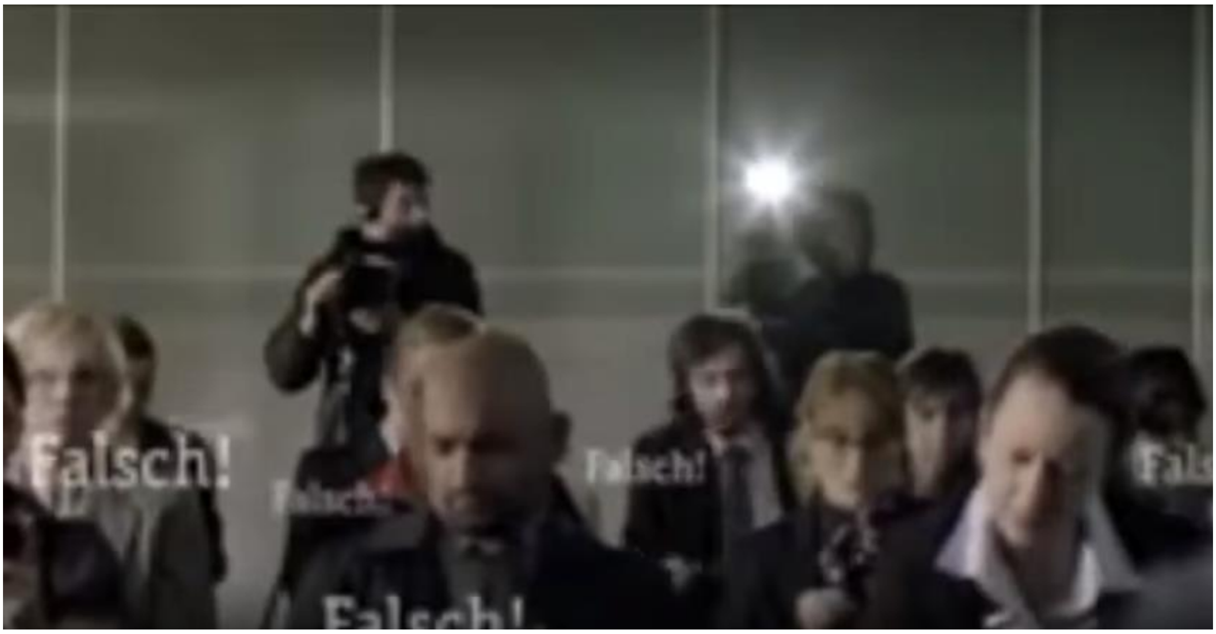


*Figure 3: Example of German-language integrated titles in* Sherlock *S01E01 (BBC/Hartswood Films, 2010–) at 05′ 50″ saying 'Wrong!'*

Subtitling is used predominantly in Scandinavian and English-speaking countries but, with internationalisation, it has also spread into countries with a tradition of dubbing; for instance, at international film festivals or scientific talks for an international audience.

### 1.2.4. Signed language interpreting (SLI)

Also known as in-vision signing or visual language interpreting, signed language interpreting (SLI) is interpreting for deaf and hearing-impaired people as well as interpreting for the Deaf, whose native language is signed language and who identify themselves as part of the Deaf culture, community, and minority (Pöchhacker 2004). SLI is used to make TV shows accessible to people with hearing disabilities or impairments. There are two kinds of hearing disabilities or impairments: congenital and acquired. Congenital refers to people who are born deaf while the acquired group of people have developed hearing loss later in life. Consequently, reading ability differs between these groups. While people with congenital hearing disabilities have a lower reading speed (often due to limited access to information and/or that reading in their native language feels like a foreign language), people with acquired hearing loss or damage are more comfortable reading subtitles at the regular reading speed since these are more legible to them than lip reading or signed language interpreting (Linde & Kay 1999: 11–12). SLI is therefore used more often by people with congenital hearing disabilities.

SLI can take place as voice-to-sign, sign-to-voice, or sign-to-sign interpreting (Pöchhacker 2004: 17–18) in simultaneous as well as consecutive modes. New technology makes signed interpreting possible in settings where the interpreter is visible on a screen through video remote interpreting although he or she is in a different location (Napier 2011: 374). This mode is often used in healthcare and medical settings to assist patients. Compared to this new interpreting mode, signed language interpreting in broadcasting settings are considered to be relatively well established. News, weather forecasts, political debates, and even radio shows are available in signed language. SLI is already renowned in academia and has existed as a profession in the USA since the 1960s (Leeson & Vermeerbergen 2010: 325). With the advent of the Internet, SLI has also gained visibility in the media since videos of enthusiastic sign language interpreters have gone viral, such as those of the then-famous ASL interpreter Amber Galloway Gallego, who specialises in interpreting hip-hop concerts, or the video of the Swedish sign language interpreter Tommy Krångh at Sweden’s Eurovision Finals. SLI can be switched on by the viewer on demand or watched online.

The fact that there is a heterogeneous audience within the d/Deaf community also affects decision-making in subtitling: adjustments to reading speed and register of language (standard vs. simple language) have to be re-considered. Now that accessibility for people with hearing disabilities has been covered, I would like to draw the reader's attention to the newer mode of AVT; namely, audio description for visually impaired people.

### 1.2.5. Audio description

While the Media for All conference[5] in 2007 dedicated an entire section to AD, the Handbook of Translation Studies (2010) does not include AD as a subject worth a whole chapter and Remael touched upon it in her article only in passing, describing it as 'an adapted aural version of subtitling' (Remael 2010: 13).

Since there were no educational guidelines on the practice of audio description (AD) or unified logo for TV programmes—in fact, no unified standards whatsoever in AD—the ADLAB (Audio Description: Lifelong Access for the Blind) was set up and funded under the Lifelong Learning Programme of the EU from 2011 to 2014 with the subsequent project entitled, ADLAB PRO, which has been running since 2016. The ADLAB project suggests the following definition for AD:

> "Audio description is the describing of film or other audiovisual products in the gaps between dialogue or other sound features for the benefit principally of the blind and sight-impaired community."
>
> (ADLAB 2014)

Chaume extends the definition by adding the dimension of language and plot and defines AD as:

> "[I]ntralingual narration that consists of decodifying images and transforming them into words, what Roman Jakobson called intersemiotic translation, and many people do not consider it to be (interlingual) *proper* translation in the strictest sense."
>
> (Chaume 2013: 111)

This AVT mode, therefore, does not count as *translation proper* (according to Jakobson as cited in Chaume 2013), but is one of the modes, among dubbing, voice-over, and fandubs, which is 'based on recording and inserting a new soundtrack and subsequent sound synchronisation' (Chaume 2013: 107).

For accessibility reasons, it is necessary that the audio descriptor sees the whole picture, does not forget what has been mentioned and what has not, and tells the visually impaired listener what is happening on screen so the listener can follow the plot. So, rather than describing the colour of the dress, it might be more important to describe the facial expression

---

[5] For more see *Media for All: Subtitling for the Deaf, Audio Description, and Sign Language*

of the person shown, and rather than talking about a general close-up of the closet, it might be more relevant for later reference to mention a green box in the closet.

AD began in 1940 on Spanish radio. In 1991, the EU set up the European project AUDETEL, which served as a basis for the Audesc system used in Spain and has subsequently influenced AD in other European countries. The first commercial programme to be audio-described on television was aired on an Andalusian autonomic channel in 1995 (Orero 2007). In the UK, the first AD took place in the mid-1980s in a family-run theatre and did not infiltrate TV until 1991 with the EU programme AUDETEL, which was responsible for the roll out of AD projects in Belgium, Portugal, and Italy as well (ibid.). In Austria, however, it was as late as 2004 before the Austrian public broadcaster ORF audio-described its first show. While for the entire year of 2009, the ORF had only audio-described 112 hours of programmes, by 2015 this number had climbed to three hours per day of AD TV programmes for selection (ORF n.d.). Audio description has gained momentum in research: even automated speech recognition (ASR), MT, and speech synthesis have been taken into consideration for this technique.[6] ASR and MT technology are also currently being tried out and used in subtitling software and is discussed in the following sections.

## 1.3. Subtitling software

Subtitling software is used on a PC or laptop to watch a digital copy of the video and to set in and out times (referred to as *spotting, cueing, time-coding* or *timing*) and horizontal positions at which the subtitles appear on screen. This software assists with certain features, which can be pre-defined or pre-determined by the customer (usually broadcasters, e-learning companies, etc.), such as maximum words per minute, minimum and maximum presence on screen, line length and minimum gaps between two successive subtitles; these are all saved in a customer template. Further features include whether the position of the subtitle should be set according to the speaker's position or always centred, whether there are colour codes for different speakers, and also how many frames should remain empty before and after a shot change. For instance, by moving the video frame by frame, it is possible to have the subtitle come in with a new scene and go out one frame prior to the next scene. There are guidelines and rules that apply in general contexts, such as the reading speed (the average number of words per minute a human can read in a specific language) and style guides that apply to specific customers.

---

[6] For more see Matamala, A. (2016). *The ALST Project: Technologies for Audio Description.*

Subtitling technology has advanced since the first programs from the 1970s and focuses exclusively on subtitling tasks. Some applications have integrated translation tools that show SL and TL in two columns next to each other at most, but these do not include translation-related features such as a translation memory or term search function. Subtitling software applications that are available for users range from freeware for amateur subtitlers and fansubbers to professional subtitling software for in-house and freelance subtitlers, to crowdsourced subtitling on online platforms or in a cloud, and—last but not least—to auto-generated, auto time-coded and even machine-translated subtitles made possible through the use of ASR. This subject is discussed in *3.2 Market analysis: Existing subtitling programs and their translation features*.

The subtitling software enables the user to use a translation template file (called a *master file*) that consists of written subtitles in the SL (usually English as a relay) with preset in and out times for a specific video. This template is then used to translate directly in the SL file into other languages. Instead of a master file, a transcript of the audio can be used that does not include in and out times and therefore requires spotting. Another option to create a subtitle file is for the subtitler to create subtitles from scratch with the only source available being the video including audio, which needs be filled with subtitles. Subtitles for d/Deaf and hard-of-hearing viewers also need to describe sounds and actors' intonations that are relevant to the plot. The source, therefore, is only the audio from the video and the target product is a written and time-coded subtitles file. In summary, there are three ways to create a translation file: use a template consisting of the SL text and preset in and out times that can be adjusted by the subtitler, translate a transcript, or translate from the audio and spot the video. The following paragraphs present an overview of the most commonly used settings for subtitle files (with no claim to be complete):

**Time** – The subtitles need to be synchronised with the dialogue and images. Some of the terms used to describe this process include time-coding, timing, cueing, or spotting. The time-codes determine the in and out time of subtitles. For accurate in and out times, it is critical that the software can move along the video frame by frame. This is a task either carried out by the person responsible for the master file (e.g. a technician) who prepares a template, which is then used by translators for all other languages, or by the subtitler for SDH or interlingual subtitles created from the video.

**Duration** – The difference in time between in and out time equals the duration. This is the parameter for how many seconds and frames a subtitle is displayed on screen. Within the subtitling software, a minimum and a maximum duration can be set (for example, a subtitle can appear for at least 1 second 0 frames and for up to 6 seconds 0 frames at the most).

**Reading speed** – The reading speed is the relation between the number of words (or characters) in the subtitle and the duration of the subtitle's presence on screen. The reading speed is therefore expressed in wpm (words per minute) or cps (characters per second). The reading speed varies from language to language and target audience to target audience. It is a widely-discussed subject in the industry with conventional values ranging from 140-180 wpm or 12-17 cps (Díaz Cintas & Orero 2010) up to a maximum of 160-180 wpm (Ofcom 2014) for English and 13-15 cps for German subtitles for hearing-impaired audiences (ZDF n.d.). Note that children and people with congenital hearing impairments have lower reading speeds and might need simpler language to follow the plot. Georgakopoulou has observed that:

> "Adult reading speed is calculated at ECI at […] approximately 180 words per minute for non-double-byte languages, whereas children's reading speed is set at considerably less and hovers between 120 and 140 words per minute."
>
> (Georgakopoulou 2009)

For more on reading speed with congenital hearing impairments see *1.2.4 Signed language interpreting (SLI)* and for more on standard values and latest studies see *2.1 Quality – guidelines and style guides.*

**Minimum gap** – To avoid a flickering effect for readers, a setting is available to insert a *minimum gap between two successive subtitles*. A gap is usually set for one to two frames. The program then observes the set rule when the user adds a new subtitle. This minimum gap has to be observed before scene changes as well as for the duration of a scene.

**Formatting** – The subtitles' formatting can be set individually for each file or with the help of a template for many files. Formatting settings can apply to font size and colour (for SDH on teletext, main speakers can have a colour assigned to them), background colour (for teletext, it is usually white font on black background), positioning (usually centred or in accordance with the speaker's position), language, automatic line breaks after a set amount of characters per line (35-43) and number of lines per subtitle (usually a maximum of two but can be as high as three to five in non-professional subtitling).

As the UK's communications regulator, the Ofcom (formerly known as ITC until December 2003) provides guidelines on the provision of television access services, checks UK broadcasters' compliance, and publishes reports on the quality of live subtitles. On the website, the following definition for subtitling is given: 'Subtitling is text on screen representing speech and sound effects that may not be audible to people with hearing impairments, synchronised as closely as possible to the sound' (Ofcom 2017).

This definition is superficial and does not elaborate on the complexity of subtitling as it provides solely a definition for so-called *closed captions* or *subtitles for deaf and hard-of-hearing people* (SDH) and neither distinguishes the different subtitle types nor the challenges faced during the process of subtitling. Thus, one might ask what the different types of subtitles are and how can they be classified.

Above, I have explained commonly used settings across all subtitling programs. Different forms of subtitles, however, require different handling and have different target audiences. The following sub-sections, therefore, cover the various subtitling forms with the specific attributes, differences, technological prerequisites, and resources used to create them.

### 1.3.1. Closed vs. open subtitles

Closed subtitles are the type of subtitles that can be switched on and off by the viewers as they like via a remote control and/or a menu on interactive platforms, such as DVD or Netflix. In the USA, the term *(closed) caption* denotes SDH, which has been in use since the mid-1970s (Díaz Cintas & Anderman 2009: 5). SDH were posed as a solution to the potential loss of advertising revenue due to 'the ten per cent of hearing viewers who reacted unfavourably to captioned television' (Linde & Kay 1999: 8). To receive closed captions or subtitles in the USA, a decoder was needed until 1993, when TV sets with built-in decoders became available; consequently, closed subtitles to this day can be switched on and off by remote control on *line 21* in North America. For Europe, *TELETEXT* was implemented in 1973 and closed captions can be accessed by entering pages 777 or 888 in most European countries (Díaz Cintas & Anderman 2009: 5). Remote controls today feature a *subtitles* button in case the subtitle signal is transmitted in a different way.

In contrast to closed subtitles, open subtitles are defined as 'text that has been inserted in the original picture by the maker of the film or programme (or a title that replaces it)' (Ivarsson 2003: online) and are 'an integral part of the film or programme and cannot be removed according to the wishes of the viewer' (ibid.). Open subtitles can appear anywhere on the screen and are usually burnt in to let the viewer know where or when the scene is happening, and whether it is a flashback or a text that has been received in the plot. Open subtitles also are shown when actors speak a foreign language to let the viewer know what the conversation is about. Additionally, subtitles in the opera or in theatres can be considered to be of the open variant.

Closed subtitles should never coincide in timing with open subtitles or cover them. In films, open subtitles are usually translated and inserted in place of the original subtitle for translated versions. If the SL is not replaced by the translation, the translation will be shown as an open subtitle in the lines for closed subtitles. These subtitles are usually inserted in the process of production and post-production via specially built software.

Closed subtitles can be either live or offline subtitles. They can also be either intralingual or interlingual or even both at the same time as, for instance, in Chinese movies for language-learners.[7] This does not always apply to open subtitles. For further classification, which mainly applies to closed subtitling, I would like to outline the differences between live and offline subtitles.

### 1.3.2. Live vs. offline subtitles

Live subtitles are created in the moment that the speaker says something in a show on air and appear on screen with a time lag or so-called *latency* of around three seconds. This latency time is recommended by Ofcom as the maximum but, in reality, it hovers around five seconds. Latency once was two to three seconds (Ivarsson & Carroll 1998: 134) when experienced typists were used to create live subtitles via a chord keyboard. The technology used today for live subtitling is speaker-dependent ASR, which is trained to recognise the voice and speech of one particular speaker (see more on this in *4.1.2 ASR*). The technique used to create live subtitles is called re-speaking. For re-speaking, two subtitlers usually sit in a booth and see and hear the broadcast on air. As there are two of them, they can take over from each other in case of technical problems or in long shows where breaks are needed. The subtitlers listen to what is spoken on air and re-speak it in the same language, while also dictating punctuation, into the ASR software, which generates written text that appears as closed subtitles on TV. The subtitles are usually formatted as white font on black background and presented in scrolling or in block form. The ASR software, however, does need to be trained on foreign words and names to process them correctly.

Live subtitles are mainly used in such scenarios as news broadcasts, weather forecasts, sports and music events commentaries and also for conferences; thus, all formats that play an important role in the accessibility to and participation in social life. Due to the importance of the subject of accessibility, guidelines and standards—particularly for live subtitles—have been set up to ensure and assess quality. These guidelines and regular reports on quality take

---

[7] Read more on www.learn-chinese-from-movies.com

into account such attributes as reading speed, presentation, and delays. In 2013, the aspect of linguistic quality was also taken into account by Ofcom for the UK. To evaluate the accuracy of re-speaking, the NER model[8] by Romero-Fresco and Martínez Pérez (2015: 28-50) is used, which is (N-E-R)/N x 100 = %, where N is *number of words* in the re-spoken text, E is *edition errors* caused by the re-speaker's omissions or paraphrasing, and R stands for *recognition errors* such as insertions, deletions, and substitutions (which are also used in the WER model applied in *4.1.3 Field test (1): YouTube's automatic caption*).

When subtitles have been prepared before the broadcast starts and are subsequently fed into the play-out system during the show, this is called semi-live subtitling (Jüngst 2010: 138). Live and semi-live subtitles do not fall under the scope of this thesis as re-speaking is a singular technique that requires its own technology. Therefore, no solution for any challenge that this subtitle form may pose will be presented.

By contrast, offline subtitles can be prepared by using a regular keyboard and also via integrated speech recognition and dictation software. They can be prepared at any time before their screening, and therefore, are mostly used for films, series, e-learning, and other videos that have no need for subtitles to be created ad hoc. The problems entailed in this subtitle form are addressed in the sub-sections below.

### 1.3.3. Intralingual vs. interlingual subtitles

According to Jakobson (1959), there are three types of translation: intralingual, interlingual, and intersemiotic translation. *Intralingual* means rewording within one language, while *interlingual* is the translation between languages. *Intersemiotic* means 'transformations in which information provided in one semiotic system (e.g., language in a novel) is rendered in another semiotic system (e.g., visual scenes in a film) either partially or completely' (Göpferich 2010: 374). Audio description is one such intersemiotic translation.

While interlingual subtitling translates both from spoken to written text (also auditory to visual) and from one language into another, intralingual subtitling is mainly used to make content accessible to d/Deaf and hard-of-hearing people as well as people for whom the language spoken in the audio is a foreign language, and include descriptions of sounds, music, and intonation. Interlingual subtitles, in contrast, do not include name tags or sound and mood description.

---

[8] Visit http://www.speedchill.com/nerstar to have the NER calculated online. NERstar is an evaluation tool funded by the European Commission and co-funded by 12 partners under the HBB4ALL project.

The process of interlingual subtitling is usually done once the whole film has been finished, including its post-production of mixing the sounds and editing. Only then, in the process of post-post-production, as Jüngst refers to it (2010: 7), does the AVT for interlingual subtitles and dubbing go into production with the transcript at hand. It is not until this process has been completed that intralingual subtitles go on order. This is because, at the point of ordering SDH, the dubbed version of the film is already available, and interlingual subtitles often do not match what is said, while intralingual do.

Offline subtitles can be either intralingual or interlingual, while live subtitles are always same-language subtitles. Since closed captions are created for hearing impaired and language-learning people, they are intralingual subtitles, but can be interlingual if also the sounds and other aural channels need to be translated.

Within this thesis, however, I use intralingual subtitling as a synonym for SDH and focus on the issues related to both intralingual and interlingual subtitles so as to propose a solution in Chapter 4. A concise overview of all translation and subtitling features needed for this solution is presented in *Table 5: Concise CAT requirements specific to subtitling forms.*

Since subtitling is one of many audiovisual translation modes, translators of interlingual and intralingual subtitles should be supported in the making of subtitles not only on a technological level, but also on a linguistic level. In the following chapter, I investigate possible arguments in favour of a supporting technology for translators in the field of subtitling.

# 2. Arguments for a CAT tool for subtitling environments

If there is evidence that subtitlers need a CAT tool, I have collected it in this chapter. To do so, I focused on data from history, academia, and current trends, and drew my conclusions from this data. Additionally, being a subtitler myself, I know the problems faced during the process of subtitling. I begin by discussing such dimensions as quality and costs, and move through history to the current trends of AVT in different countries to substantiate my hypothesis.

## 2.1. Quality – guidelines and style guides

So, what is a TM primarily good for? My hypothesis is that components such as a TM, a term base, a thesaurus, and automated features like MT are vital tools to improve consistency—and by that means—quality. But how is quality measured in subtitling? There are numerous guidelines, norms, and rules for subtitling, which vary from language to language and country to country; nonetheless, they share certain elements. The wide range of guidelines in force proves the importance and visibility of subtitling in media today.

From public broadcasters, such as UK's BBC, Austria's ORF, and Germany's ARD, through to globally operating platforms like Netflix, all have either their own subtitling guidelines or comply with regulations imposed upon them by regulatory bodies, such as Ofcom's *Code on Television Access Services* for the UK's public broadcasting stations. Private broadcasters have also been obliged to observe captioning standards; for instance, Canadian private broadcasters must comply with CAB's (The Canadian Association of Broadcasters) standards, which were set up to ensure 'the provision of high quality closed captioning' (CAB 2016: 7).

Institutions and associations have attempted to establish standards, raise awareness for price-dumping, unify rules for professional subtitling, and even create trade unions in their respective areas to ensure high quality and proficiency in subtitling. Such associations include ATAA for France, ESIST (European Association for Studies in Screen Translation) with their condensed two-page *Code of Good Subtitling Practice*[9] by Ivarsson and Carroll (derived from *The Principles of Subtitling* in 1998: 63-78) and AVTE (AudioVisual Translators Europe) for Europe, NAVIO for Norway (Norsk audiovisuell oversetterforening), and SUBTLE for the UK, among others.

Some of the regulatory bodies within the European Union have been set up as a reaction to the EU Directive 2010/13/EU, which is the latest amendment to the Council Directive 89/552/EEC, also known as the Audiovisual Media Services Directive, which states that '[p]roviders are obliged to improve the accessibility of their services for people with a visual or hearing disability' (AMS Directive Summary). The Directive emphasises the right to be part of social life and states in item 46 that:

> "The right of persons with a disability and of the elderly to participate and be integrated in the social and cultural life of the Union is inextricably linked to the provision of accessible audiovisual media services. The means to achieve accessibility should include, but need not be limited to, sign language, subtitling, audio-description and easily understandable menu navigation."
>
> (Directive 2010/13/EU)

**Regulatory bodies** – As one regulatory body, the British Ofcom (previously ITC) not only provides guidelines, but also monitors compliance of British broadcasters. Up until 2013,

---

[9] For more see https://www.esist.org/wp-content/uploads/2016/06/Code-of-Good-Subtitling-Practice.PDF.pdf

Ofcom used to check only the amount of provided subtitles that were reported to it by the broadcasters, but not their quality with the result that many complaints were submitted (The Guardian 2015: online). To tackle these problems, from 2013–15, Ofcom conducted six-monthly quality checks of live subtitles, taking into consideration such factors as accuracy, latency, and speed. The fourth and last report had a NER result for average accuracy of 98.55%, an average latency of 5.6 seconds (higher than the recommended three seconds), and, apart from 92% of broadcasts having short passages of 200 and more wpm, an average subtitling speed of 160-180 wpm, which is the recommended speed (Ofcom 2015, November: 13).

The Ofcom, however, is not only the UK's communications regulator for television, but also for radio, broadband, and postal services. Moreover, Ofcom is accountable to parliament and operates particularly under the Communications Act 2003, which states that 'Ofcom's principal duty is to further the interests of citizens and of consumers, where appropriate, by promoting competition' (Ofcom n.d.: online). Ofcom's *Code on Television Access Services* defines the requirements for broadcasting stations regarding accessibility in TV services and provides guidance on the best practice for subtitling, audio description, and signed language interpreting (Ofcom 2015: 2). In Annex 1, it is stated for BBC that 100% of content should be subtitled, 10% audio-described, and 5% signed, for ITV1 and Channel 4, the percentages are 90%, 5%, and 10% respectively; for Five and S4C1, these percentages should be 80%, 5%, and 10% respectively (ibid.: 11).

In Ofcom's best practice for subtitling (Annex 4, Ofcom 2015: 18-19), it is stated that subtitles should be presented within the *safe caption area* of a 14:9 display in a clear way and without obscuring the mouth or face (A4.14). They should contain no more than one sentence and comprise of two lines at most (A4.16). It is further recommended to colour-code speakers, mark pauses, use italics for emphasis and describe the mood of the music (A4.17). The subtitles should always be synchronised with speech, have a maximum delay of three seconds for live subtitles, and not run over shot changes (A4.18). Regarding speed, a rate of 160-180 wpm is advised with the recommendation of a 'slower speed […] for young children' (A4.19; ibid.: 19). Furthermore, Ofcom guidelines encourage accurately edited and reviewed subtitles (A4.20).

As for reading speed, more recent studies were conducted by the BBC research and development department (BBC R&D), which concluded that viewers rated subtitles as too slow when the subtitles were not synchronised with the speech due to a time lag in live subtitles. BBC R&D stated in their summary that:

> "In conclusion, this study, along with previous work by BBC R&D on the effect of errors and lag on enjoyment of subtitles, has shown that guidelines should not recommend subtitlers edit subtitles to a specific rate. Doing so will likely reduce viewer enjoyment of the content. Instead, we recommend that subtitlers should aim for well-timed, accurate, verbatim subtitles to provide the best audience experience for subtitle users."
>
> (BBC R&D/Sandford 2016: online)

**Broadcasters** – The public broadcasters ARD and ZDF (Germany) in collaboration with the SFR (Switzerland) and ORF (Austria) agreed on an outline of *Untertitel-Standards* to make their content accessible in a unified way; although, each station has the freedom to create their own style guides regarding editing and layout.

Presentation is set at a maximum of two centred lines at the bottom of the screen without scrolling. Font size is Videotext with a maximum length of 37 characters per line. Minimum duration is one second, reading speed is to be set to 13-15 cps, lower for broadcasts for children, minimum gap of one frame. Colour-coding and speaker tagging is compulsory. Inserts should not be covered. All subtitles should be synchronised with speech in concord with shot changes. Furthermore, the standard demands that the German spelling rules be observed, that no information already conveyed by images be subtitled, that no more information be given to the users of subtitles than hearing audiences have (i.e. colour-coding the murderer before s/he is revealed), and that music with its artist, song title, and lyrics be denoted (ZDF n.d.).

Such rules and best practice examples make clear that, although the specifically technical aspects of subtitling processes are accounted for, they do not give guidance on translation. This is due to the fact that all these guidelines regulate subtitling in compliance with the EU Directive for accessibility in audiovisual media, therefore, these standards apply to SDH and not to interlingual subtitles.

**Academia and associations** – Ivarsson and Carroll's *Code of Good Subtitling Practice* (1998: 157), however, shows more precise instructions on how to handle the linguistic aspects of subtitles and how to translate them. Here, they have pointed out that '[t]ranslation quality must be high with due consideration of all idiomatic and cultural nuances' (ibid.). Furthermore, coherence, semantic units, syntax, register, grammar, and 'close correlation between film dialogue and subtitle content' (ibid.: 158) are addressed. Intriguingly, it is recommended that names and interjections be subtitled for d/Deaf and hearing-impaired viewers, as opposed to the commonly recommended way of omitting them as 'dispensable elements' (Kovačič 1991: 409) due to space constraints: repetitions, however, can be left out (ibid.).

In the same year that Ivarsson and Carroll wrote their *Code*, Karamitroglou made an attempt at creating ‘A Proposed Set of Subtitling Standards in Europe’ (1998) that contained a section on ‘target text editing’, which not only included rules on omitting, retaining, and altering linguistic items, but also on the possible handling of dialects, taboo words, and culture-specific linguistic elements. Despite the fact that these rules and standards are useful in interlingual subtitling, they have not been in force since their creation, especially throughout Europe, where countries such as Poland, Hungary (and others) have not yet applied any of the proposed international rules. These findings must be taken into account in the process of designing a CAT tool for subtitling environments to pose a solution to the issue of having various customers with various style guides.

For instance, one subtitle that adheres to one guideline or style guide, might not do so for another customer. What happens when such a subtitle is stored in a TM and does not comply with the guidelines of another company? There may be certain rules that are accepted throughout the whole industry like wpm and the maximum time on screen, but other rules concerning layout and the handling of dialects might be different. Then again, as we have seen, wpm rates vary from language to language as well. Consequently, this issue is taken up further in my analysis.

This section demonstrated that a TM is needed; yet, it also showed that there are even more problems to be tackled on the way to designing a well-functioning solution. Guidelines show the importance of subtitling itself, but does the existence of guidelines relate to the importance of a TM as well? Style guides, however, are a vital means for reference for translators who work in subtitling. There are not only style guides about the style and layout of subtitles, but also reference files with (for example) names and colour-coding of the main characters, in and out times for theme songs and their lyrics that need to be consistent throughout all files so so each episode of a show remains consistent and familiar to its viewers. These reference files and the fact that many companies use different style guides have led me to the idea of setting up templates that would save time, raise productivity, observe quality, and thus, save costs for translators (rather than for companies).

## 2.2. Lower costs, higher productivity and quality

‘There is a clear need to increase the productivity of subtitle translation procedures, reducing costs and turnaround times while enhancing the quality of the translation results’ (CORDIS 2014) states the SUMAT project in its objective funded by the EU through the years 2011–14.

It is indeed true that there is the argument of a much-needed cost reduction in postproduction, which I will elaborate on in this section. SUMAT's suggestion, in contrast to the proposal in this paper of using a CAT tool, was to use MT to speed up processes and thereby increase productivity in subtitling. (See more on this project in *3.2 Market analysis: Existing subtitling programs and their translation features*.)

Citing previous studies, Declercq has stated that 'using the TM even slowed down productivity by 2.5 per cent, whereas MT increased this by 24.5 per cent, a combined difference of 27 per cent or nearly a third' (2015: 486). I could not verify these figures as no resource is named, but I found studies with slightly different outcomes. These studies were conducted by Bowker in 2005 on quality and productivity of translation memory systems with students as translators (Bowker 2005: 13) and by Guerberof Arenas with the results published in 2008 on and in *Productivity and quality in the post-editing of outputs from translation memories and machine translation,* conducted with professional translators. As regards translators' **processing speed**, the results show that MT is about 16 per cent faster than human translation and 5 per cent faster than the use of a TM (Guerberof Arenas 2008: 34), whereas the **gain in productivity** when using MT and post-editing is between 13 and 25 per cent, while when using a TM, it is 10–18 per cent as compared to translating new segments (2008: 37, 65). However, Guerberof also draws attention to the results after allowing for error correction, which showed a 9 to 25 per cent productivity gain for MT in contrast to a loss of 3 to 8 per cent for TM (2008: 65), and she also has noted the fact that 'productivity seems to be subject dependant [sic!]" (ibid.: 38). These low results for TM are due to quality of translation outcomes. As for **quality**, an error analysis was conducted to compare the number of errors per type of segment: they amounted to a total of 65 errors in TM segments, 34 in MT segments and 27 in new segments translated by human translators (ibid.: 41). The high number in TM segments is explained, inter alia, by the fact that TMs reproduce errors by reusing the same segment without reassessing it (ibid.: 43) and 'translators using TMs may not be critical enough of the proposals offered by the system' (Bowker 2005: 19). In a study six years later, however, Guerberof states that in 'translators […] using machine-translated output […] productivity and quality is not significantly different from the values obtained when processing fuzzy matches from translation memories in the range 85–94 %' (Guerberof Arenas 2014: 165). And these two factors are what counts for my thesis: quality and productivity. To support these two factors, a combined use of TM and MT for the future AVTEnT could be considered, but let's look further now at how costs are handled at present.

Within the DVD and TV subtitling industry, the following solution was proposed to reduce costs in subtitle production on the one hand and to simplify management of subtitle files on the other hand: a time-coded universal subtitle file template in English (Georgakopoulou 2006). It is also referred to in the industry as the *master file*, which serves as a template for all languages that require translation. The cueing task is therefore carried out solely for the English file, which is then reused as a template by all other translators. This saves time and costs since only a few changes (Carroll 2004)— or even none at all—are made to the in and out times, and only one person needs to perform and get paid for the task of cueing, rather than each translator for their respective language. On the face of it, this sounds like an excellent solution, but as the data from the guidelines in Section 2.1. above shows, each language has its own length and thus its own wpm rate, meaning that preset time codes for an English file cannot suit the time needed for German subtitles, which are on average 20 per cent longer in number of characters. For the time being and in the absence of any other solution, this is an easy workaround, but it also limits translators in their choice of words since they are bound to the preset in and out times and usually do not get the chance to make any time adjustments. Translators in this scenario are to carry out translator tasks while someone else carries out subtitling tasks, and the profession of interlingual subtitlers seems to be wiped from the list of language professions. The sort of template file described above is also used in the localization of video games from e.g. Japanese to FIGS (French, Italian, German, Spanish) for the European market, where quick turnovers are required to meet the schedule for the release date in different countries.

In light of the above, the question arises: if there is a solution at hand, would a TM help subtitlers to work faster and therefore earn more money while at the same time maintaining excellence? What do subtitlers and translators think? When I was looking for an answer on the internet, I found a few discussion threads in forums on this subject and collated them in Table 1:

|  | PROZ.COM[10] (Sept. 2004) | PROZ.COM[11] (March 2008) | PROZ.COM [12] (March 2016) | TOTAL | ARGUMENT |
|---|---|---|---|---|---|
| **Useful** | 2 | 1 | 2 | 5 | Repetitive content, re-use of good previous solutions, boost speed |
| **Not useful** | 3 | 0 | 2 | 5 | Artistic translation, space & tone can differ, repetitions unlikely, watching video is vital |
| **Neutral** | 1 | 2 | 0 | 3 | CAT can be used for text |
| **Total** | 6 | 3 | 4 | **13** |  |

*Table 1: Internet discussions on whether a CAT tool in subtitling is useful*

As shown above, five people argued in favour of and five against the use of CAT tools in subtitling, although obviously, this is not a representative sample. On ProZ.com, many different topics can be addressed by professional translators around the globe, yet I could only find three threads on the topic in question. The discussions show that a CAT tool is not at all frequently asked for, but some consider it useful in a range of cases, mainly for shows and videos which contain repetition, but also to boost speed because good previous solutions can be reused. Others, who argued against it, did so mainly because subtitling is regarded as artistic and comparable to literary translation, and therefore the use of repetitions would be unlikely since tone and register can differ from one film to another. That argument may be valid when applied to the subtitling of films and series, but it does not take into account subtitling for marketing and commercial purposes, as well as scientific contents, such as e-learning, which often use rather neutral language. Furthermore, one critical argument in the discussions was that watching the video was an integral part of subtitling and switching from a CAT tool to the video was too time-consuming, although this is what subtitling used to be like in the 1990s with VHS, one computer and one TV set. The feature of playing videos in CAT tools is supported by Transit NXT, which I scrutinise in *3.3.2 STAR Transit NXT*. Nowadays, subtitles are in use in any audiovisual context and can be created anywhere the translator is located since the user needs only one laptop. Just think of the button on your TV remote control (Díaz Cintas 2015: 634) or of YouTube, where you can click on the cogwheel to get fully automatically generated real-time and even auto-translated subtitles.

---

[10] http://www.proz.com/forum/subtitling/24726-cat_tools_for_subtitling.html

[11] http://www.proz.com/forum/subtitling/100862-subtitling_and_trados.html

[12] http://www.proz.com/forum/subtitling/299030-are_there_any_cat_tools_for_subtitling.html

Many issues that I already discussed in *3.1 Forms of subtitling and their technological needs* have been revisited here, from a different perspective. The learnings I have derived from this chapter are that I now know that: a) the idea of a CAT tool for subtitling environments has been acknowledged within professional translation and subtitling; b) that a TM can raise productivity, but due to error propagation can also lower it; and c) that a combination of a TM and MT would be a good way to go. What other facts could argue in favour of a CAT tool in subtitling processes? I will take a look at what history can tell us and what the future is set to bring.

## 2.3. 1920 to 2020

In the 1920s, movies developed from silent movies with so-called *intertitles* (c. 1903) and from 1909 onwards, *sub-titles* (see Ivarsson 2004: online), to movies with sound called *talking pictures* or *talkies* (c. 1927). Since the movies were shot in English and, compared to today, only a small percentage of the world's population spoke English, some experiments with possible translation modes were made. Not only did the early talkies try to avoid the intertitles that were used in silent movies, they also experimented with re-shooting films. For instance, the *Laurel and Hardy* films were shot in English and re-shot with similar plots and scenes by the same actors but in many different languages. In this case, the actors who played Laurel and Hardy and who were English native speakers, had to re-shoot their lines in German, Italian, Spanish, and French. This solution remained unsatisfactory and so investigation into other distribution modes was conducted. Subsequently, dubbing and subtitling became the most commonly used options. Since subtitling 'is comparatively cheap (subtitling only costs between a tenth and a twentieth of a dubbing), it became the preferred method in the smaller language areas, such as the Netherlands and the Scandinavian countries', states Ivarsson (2004: online). In addition to these countries, Finland, Greece, and Portugal also have had a strong tradition of subtitling. Other countries, however, such as France, Italy, Spain, and Germany became dissatisfied with subtitling and instead started to dub films, which since has become the main AVT mode to date (Tveit 2009; Ivarsson 2004). Jüngst has claimed that the reason for this development is due to the fascist totalitarian history of those countries. Propaganda was easier to distribute when the original could be edited out and superimposed with untruths in the national language; although, she does add that this works out just as well with subtitles—as Portugal demonstrated (Jüngst 2010: 4). Yet, subtitling had not become obsolete; indeed, quite the opposite happened with the appearance of national guidelines and EU directives for d/Deaf and hard-of-hearing audiences.

The studies, books, and documentation on subtitling technology and computer-aided translation technology that were most helpful during my research were written at the end of the 1990s and in the early 2000s. However, barely any progress or further development in the basic functions of either technology has been achieved since then even though technology and digitalisation have grown rapidly. Díaz Cintas has noted that ‘the real impact came about with the start of the digital revolution in the 1980s. Boosted by vast improvements in computing technology, it marked the beginning of the information age and globalisation trends’ (2015: 632). I agree partly to this statement as there have been a few changes to processes, but these changes are only due to the technical development of the medium itself rather than the result of professional needs. Digitalisation caused the medium to change, and with it, the processes of subtitling changed. Such developments in subtitling that are worth mentioning are outlined below.

In the 1990s, translators working in the field of subtitling were given videotapes with magnetic time-codes on them and had to replay the videotape for editing and checking; nowadays, broadcasting companies provide the videos in MPEG format and expect the subtitles to be electronically delivered in an STL file.

Ivarsson and Carroll pointed out in 1998 that live subtitlers were required to have high typing speed skills and needed at least a year’s training on using a chord keyboard and ‘highly developed abbreviation programs, which [could] increase typing speeds by 20-40%’ (1998: 134); nowadays, live subtitlers are trained in re-speaking into a speech recognition software tailored to their vocal patterns.

In February 1991, the European Broadcasting Union (EBU) in Switzerland set up a standard data file format to make an exchange of subtitles feasible, called the EBU subtitling data exchange format STL. In their documentation, they named the medium for exchange, which was ‘a 3.5-inch high-density portable magnetic disk (microfloppy)’ (EBU 1991: 3). Floppy disks, later CDs, then to be followed by DVDs have all since been replaced by USB sticks and online cloud storages, FTP servers as well as email attachments. The file format STL was kept as a standard exchange format, but had a follow-up in 2012: the EBU-TT (EBU Timed Text) subtitling format.

The previous developments show that analogue data carriers have been replaced by digital and electronic media, which ‘opened up new avenues not only for the production but also for the distribution, commercialization and enjoyment of subtitles’ (Díaz Cintas 2015:

632). VHS cassettes have been superseded by DVDs, chord keyboards have been superseded by speech recognition software, and antenna transmission was replaced by Digital Video Broadcasting – Terrestrial (DVB-T), which has recently been superseded by DVB-T2 and Hybrid broadcast broadband TV (HbbTV) with content from the internet.

With the growing impact of the internet, the generation of data has also grown exponentially; for example, if you were to express information from all books and historical documents in gigabytes (GB), you would see that the world has generated a data volume of 2 exabytes (2 EB = 2 billion GB) up to the year 2000. Today, 2.5 exabytes are produced *per day* (Khoso 2016) by (among other things) taking and uploading pictures, videos, tweeting and re-tweeting, and online shopping. The market research company IDC (International Data Corporation) in cooperation with Dell EMC has estimated that ‘by 2020 the digital universe – the data we create and copy annually – will reach 44 zettabytes, or 44 trillion gigabytes’ (IDC 2014).

To clarify the fast developments that I discuss here, we can take YouTube as a case study. YouTube’s inception was in February 2005 with the first video uploaded on April 25, 2005. By 2010, ‘twenty hours of video [were] uploaded to YouTube every minute’ (YouTube 2010). In 2015, the Google public policy manager Verity Harding stated at a European Parliament meeting that the number of hours of new videos uploaded to YouTube had climbed to 300 hours per minute (The Guardian 2015). With more than one billion people using this media channel (out of a total three billion people on the internet) a daily video view rate of four billion is generated. An enormous amount of traffic is caused by just this one channel, which has local versions in about 88 countries (YouTube 2017). With the growing use and development of mass communication, audiovisual translation has also gained importance and visibility, which the €80 billion funding for the European Commission’s EU Framework Programme for Research and Innovation Horizon 2020[13] proves.

These developments can be seen everywhere. For instance: with the implementation of a silent auto-play function for videos in the Facebook feed, users can watch videos virtually everywhere where there is internet access, even without the need to press the *play* button. When the surroundings are too noisy or require silence, users no longer need to forgo what is happening in the world: thanks to captioning of silent auto-played videos. Apart from the traditional subtitle carriers like DVDs and TV shows, subtitling has become ever more important in the field of marketing, commercials, e-learning, scientific content, and language

---

[13] See https://ec.europa.eu/programmes/horizon2020/ for more information on projects funded under this programme

teaching content. Its impact shows its effect not only in the industry, but also in the academic world. Following the increasing number of articles about AVT in journals, handbooks, and conference proceedings over the past decades, a specialised journal is now forthcoming, the Journal of Audiovisual Translation (JAT), which will be published in cooperation with the association ESIST. Furthermore, the book *Fast-Forwarding with Audiovisual Translation* edited by Díaz Cintas and Nikolic will be published in December 2017 with educational, professional, and academic perspectives on AVT approaches.

Regardless of the importance of AVT in the industry and academia, not to mention its low cost in production, it becomes obvious that subtitling is not always the preferred AVT mode, and each mode is not equally in use from country to country. Nonetheless, a rough overview of AVT mode trends may offer a clearer view on whether a CAT tool or elements thereof could help professional translators and SDH subtitlers.

## 2.4. Trends of AVT by country

Looking at studies on AVT in the past decades, it becomes clear that one country's habits concerning the preferred AVT mode differ from those of another country. To point out the importance of subtitles—not only for people with hearing impairments—I conducted some research and was about to compile the findings on preferred AVT modes in selected countries when I realised that it would not lead me to an answer because each country (as shown in *1.2 Audiovisual translation (AVT)*) has its own traditional mode while, at the same time, another mode is becoming more dominant. For example, countries with a tradition of dubbing like Spain, Austria, Hungary, Germany, France, Italy, Japan, and China nowadays provide more and more subtitles, especially for international film festivals. Meanwhile, countries with a tradition of subtitling, such as Portugal, Denmark, and Norway and such countries as Russia and Poland with a strong tradition of voiceover, are also currently shifting towards dubbing with foreign productions (Chaume 2013: 108). A growing population with access to the internet and freedom of movement makes it possible for people to move across the globe where they need to learn the country's language: this also creates a demand for foreign-language subtitles. The industry for subtitles has recently outgrown films and DVDs and now embraces commercials, YouTube videos, e-learning courses, conferences, and many other areas: thus, becoming the centre of today's attention. Some media even carry multilingual subtitles; for instance, in Finland with Finnish and Swedish, in Belgium with Flemish and French, and in Israel with Hebrew and Arabic (Gottlieb 2002: 197).

Additionally, with a growing and eldering population, more people with hearing disabilities need access to audiovisual content. While Britain's BBC subtitles 100% of its broadcasts, the German broadcasting station ARD states that Das Erste subtitles 95% (Das Erste 2016) and its channel NDR subtitles 77% (NDR 2016) of their programmes. As of 2013, the Austrian broadcasting corporation ORF offered 63% (ANED 2013) of its broadcasts with subtitles. Although other forms of AVT for accessibility reasons are acknowledged in the industry and have enjoyed a rise in media, the numbers of (for example) audio-described or signed programmes are still in the early stages of development, which means that there is still much to explore.

For now, however, I would like to focus on analysing and researching possible translation-assisting solutions for subtitling environments. After considering the pros and cons of a CAT tool in subtitling processes with the result that the pros outweigh the cons, my next step in the following chapter is to find out whether R&D departments of software developing companies in the subtitling industry have come to the same conclusion as I and, perhaps, have already presented a solution to my question.

# 3. Analysis and field testing of CAT and subtitling tools

Although interlingual translation has taken place in subtitling for a longer period of time than intralingual translation for SDH (see 2.3 on the history of subtitling) and despite the ever-growing translation need for both modalities, subtitling software is, to date, designed to serve pure subtitling processes rather than to meet translators' needs.

Carroll concluded in 2004, only six years after *Subtitling* was published, that '[a]ll types of subtitling would benefit from enhanced software solutions, ranging from integrated CAT (computer-assisted translation) tools to automatic voice and cut recognition, which are starting to appear' (2004: 3). While, in 1998, Ivarsson and Carroll described a subtitler's materials as 'a script or dialogue list […] and a copy of the film (usually a recording on VHS or S-VHS video cassette or audiovisual hard disk)' (1998: 79), the materials nowadays are usually digital, and the technology used to create subtitles extend from freeware to professional subtitling software, to subtitling on online platforms or in a cloud, to auto-generated subtitles, such as those made accessible by Google's ASR on YouTube, which can even be auto-translated within the same application. In the course of writing this thesis, I apply the resources that I deem useful for the purpose of designing a CAT tool for subtitling environments and document the outcome.

In order for this thesis to be legitimate, one needs to ask the most prevailing question: Is it necessary for a subtitling software to store translation units at all? If so, why is a subtitling software including CAT tool elements still not on the market? Perhaps the need is not great enough to warrant R&D departments investing their time into such a question. Consequently, it is important to analyse the issue by looking at both the history of subtitling and the various preferred AVT modes in different countries. Further questions one might ask include whether subtitling is a discipline that is too young to make progress, or why countries simply do not use subtitling, let alone subtitling for the deaf and hard-of-hearing people but instead use dubbing or voiceover only. After Carroll had done in 2004, Díaz Cintas demanded in 2010 and again in 2015 that there be ‘translation memory systems and automated translation in subtitling’ (2010: 348). Thirteen years have passed since Carroll’s article and much has been done to work towards this goal; yet, there is very little commercially available software that features a TM to date. Thankfully, much research has been carried out that might lead to an answer as to why the market is still missing an AVTEnT, or at least, enable me to deliver hard facts that substantiate my hypothesis; namely, that, indeed, there is a need for a CAT tool in subtitling software after all. The latest statement that affirms my hypothesis that a TM in subtitling is needed is the following:

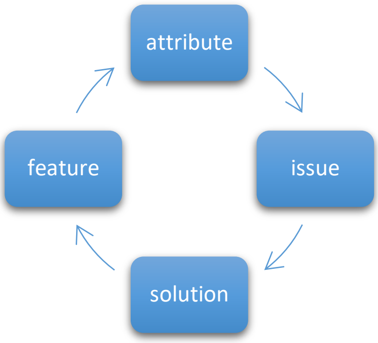
> “Yet, and perhaps rather surprisingly when compared with other areas in translation (O’Hagan 2013), little attention has been paid so far to the role that computer-aided translation (CAT) tools can play in subtitling or to the potential that translation memories and machine translation can yield in this field, although the situation is changing rapidly.”
>
> (Díaz Cintas 2015: 633)

I would like to find out the extent to which the situation is ‘changing rapidly’ and see which applications are available for translators in subtitling environments. Therefore, I delve deeper into this subject in the following sections by investigating technology that I deem considerable and useful for an AVTEnT. To explore which features may be useful, further thought must be given to the differences in subtitle forms and their specific technological needs.

## 3.1. Forms of subtitling and their technological needs

As discussed in Sub-sections 1.3.1 to 1.3.3, there are different forms of subtitles that require different ways of handling and creating them; hence, different software features. Each subtitle form entails its own difficulty regarding its translation or transcription.

Therefore, I begin my analysis by identifying translation issues in each subtitle form. I then move on to structuring them and posing solutions that are applicable in a software feature. I have therefore created the following solution-oriented map, which relates the prerequisites of a subtitle (the subtitle's attributes and needs) to possible resources in a CAT tool (functionalities of a CAT tool that can serve as a resource for the creation of subtitles) for intralingual, interlingual subtitles, and common properties. So, briefly worded, we have the following top-down analysis to create a systematic cycle:



A subtitle's *attribute* leads to a translation *issue*, which needs a *solution*. The solution is posed through the *feature*/function of a program, which shall support the unimpeded feasibility of the *attribute*.

*Figure 4: Solution-oriented cycle to map needed function in an AVTEnT*

I shall begin by applying this solution-oriented cycle to intralingual subtitling firstly by outlining the attributes and issues, which will be followed by a solution and the function resulting from it.

### 3.1.1. Intralingual subtitling

Usually, a digital copy of the video is available but no transcript; therefore, spotting and splitting is needed, which is time-consuming, especially when bearing in mind the various minimum gap and shot change rules per customer (see row (a) in Table 2). Furthermore, due to the lack of a transcript, the subtitler needs to research the music with its artist, song titles and lyrics, and additionally spot the intro or theme song of each episode (see row (b) in Table 2).

Since not only subtitling of speech and utterances is needed, but also the tagging of intonation and meaningful, mood conveying sounds and music, and voices in the off-scene, all while observing space and time constraints, concise use of language that still describes mood is essential (see row (c)).

Lower reading speeds in children and people with disabilities demand longer subtitle presence on screen, thus, lower wpm rates. To comply with the limited space and time available, certain utterances like names in the appellative mode, interjections, stuttering, pauses in speech, and other components can be omitted or paraphrased (see row (d) in Table 2).

Name tagging and colour-coding to mark different speakers are necessary to make speaker changes easier for non-hearing audiences to follow (see row (e) in Table 2). Series usually open with a recap of what happened in previous episodes or seasons, which contains condensed and quick review of dialogue and plot (see row (f)).

The intralingual subtitles should not contain more information than what a hearing audience has access to; thus, if a foreign language is spoken in the scenes without open subtitles, the foreign language should not be translated (see row (g) in Table 2). Additionally, the handling of dialectal phrases can be different, meaning that the subtitler should either tag the utterance and replace the expressions with standard language variants, or subtitle verbatim; with this solution, however, the problem occurs that dialects usually do not have spelling rules and therefore cannot follow them (see row (h)).

Having the above collated attributes identified, the following table of intralingual subtitles' issues and their respective function in a software can be mapped out:

| ATTRIBUTE | ISSUE | SOLUTION | FEATURE |
|---|---|---|---|
| **(a) no transcript available** | spotting and splitting is time-consuming | transcribe, time-code, split according to line restrictions and minimum gap | ASR for transcription and auto-cueing, after identifying shot changes, split subtitles by sentence complying with minimum gap rules and wpm rate |
| **(b) music: artist and song title must be credited** | leaving application to do research takes time | look up on web without leaving the application, declare, and time-code | ASR/Music identifier which identifies artist and title and inserts result in a format as per customer |
| **(b) lyrics search + marking as lyrics by use of colour** | leaving application to do research plus colour-coding takes time | look up on web without leaving the application, transcribe, and colour-code | ASR/Music identifier which identifies artist and title and inserts lyrics in the format as per customer |

| | | | |
|---|---|---|---|
| **(b) intro/theme song** | leaving application for reference file or to do research plus cueing is time-consuming | look up in reference file and transcribe according to timing rules | TM to identify intro from auto-transcription<br>or<br>term base to look up in template (e.g., stored within the application), spot and transcribe the theme song for each episode in a unified way |
| **(c) include description of sounds** | limited space allows only for concise language | omit, paraphrase, condense, summarise | Thesaurus to auto-recognise words and their length and auto-suggest synonyms or alternatives |
| **(d) lower reading speeds in d/deaf people and children** | presence on screen is longer, therefore lower wpm/cps rates | • omit, paraphrase, condense, summarise<br>• auto-set wpm to a lower value | • Thesaurus to auto-recognise words and their length and auto-suggest synonyms or alternatives, such as abbreviations; also identify interjections, names, etc.<br>• TM proposes approved TUs with TL specific wpm rates |
| **(e) name tagging in the off and colour-coding** | • time-consuming for translator<br>• less space due to name tags | identify and colour-code different speakers or insert names | • Automated colour-coding thanks to voice recognition and auto-insertion of either tags or colours<br>• Thesaurus for suggestion |
| **(f) opening scene with condensed recap of what** | • need to subtitle dialogue that was subtitled before | • avoid redundancy by identification of previously | • TM identifies from auto-transcription that this segment has a match |

|  |  |  |  |
|---|---|---|---|
| **previously happened** | • to then edit down according to lower wpm rates | subtitled segments<br>• and suggestion of alternative wording | • Thesaurus suggests synonyms and identifies parts that can be omitted |
| **(g) tag foreign language utterance but do not cover possible open subtitles** | overlook open subtitles, look up how to handle foreign utterance | identify open subtitles, tag foreign utterance if applicable | Possibly, OCR or auto-identification of open subtitles when text appears on screen (limit to clearly visible letters,<br>so no background letters get subtitled),<br>set rules for foreign-language tagging in a template within the application |
| **(h) dialectal phrases vs. standard language** | as per customer, sometimes tagging and standard language, sometimes verbatim subtitling is needed | save rules to a customer template and point out dialectal phrases | Term base for dialectal phrases, training for ASR in case it does not recognise them, identify and tag dialect if needed, store rules in customer specific template |

*Table 2: Top-down analysis to map intralingual subtitle needs to possible CAT features*

Table 2 shows that many issues have been identified in this sub-section, and the collection certainly has room for further development. These issues can be solved by integrating a tool that assists subtitlers in both their linguistic and technical work, such as ASR for transcription, a TM for redundancies, music identifier for songs and lyrics, term bases for several rules, and perhaps most important in intralingual subtitling, a thesaurus for auto-suggestions of shorter alternatives and abbreviations. Now, let us see whether the same functions, or other functions, are applicable to interlingual subtitling.

### 3.1.2. Interlingual subtitling

Interlingual subtitles are usually created at the same time as the dubbed version goes into production. The problem with *interlingual* subtitling is that the subtitle translators receive the transcript of the original dialogue, while the same transcript goes out to dialogue translators and the dubbing director who then adjust and change the dialogue so it synchronises with the lip movements of the SL. As a result, interlingual subtitles often do not pair up with the spoken dialogue. See Figure 5 below for an example from the Netflix Original series, *You Me Her* S02E01 (Audience Network 2016–) written by John Scott Shepherd.

| | **US sound and CC** | **Dubbed DE version** | **Interlingual DE subtitles** |
|---|---|---|---|
| **[Lori]** | Oh, good Lord. Ladies, what is the actual objective here? | Oh großer Gott, Ladies! Was soll der Aufzug bewirken? | Meine Güte, Mädels, was soll das denn darstellen? |
| **[Ava]** | Post-post-post-feminist empowerment. | Post-post-post-feministische Emanzipation. | Post-post-postfeministische Frauenbewegung. |

*Figure 5: Example of interlingual vs. intralingual vs. dubbed dialogue from* You Me Her *S02E01 (Audience Network, 2016–) at 6'07"*

This example clearly shows that the translations for dubbing and those for the interlingual subtitles were made from a different source due to processes in the post-production that need to be performed simultaneously. Since this thesis does not focus on processes within the industry, I do not propose a solution to this problem; nonetheless, I need to point out that there is an issue if the subtitles are not even close to matching the audio for people such as language-learners who, in the absence of intralingual subtitles, watch a show with interlingual subtitles to understand what is said. But since interlingual subtitles are not suitable for a non-hearing audience, the target group is hearing audiences who speak a different language than the source. The audience, in this case, perceives the SL audio, but reads the TL subtitles, and only time and space constraints pose a problem in this setting.

The problem with *interlingual translation of SDH* is that since the translators receive a transcript for translation, including a copy of the video for context, they must consider both the visual and oral channels plus the proposed SL sound description, which may be associated with a different emotion in their TL. The translator then needs to decide for the viewer which emotion the intonation carries and re-tag it while observing space restrictions.

If the translator works from a master file or template—for instance, from English into German, which results in about 20% longer TL text than the SL—editing down becomes an inevitable task, especially when taking into account the different wpm rates for each language. English has a suggested reading speed of 180 wpm, while the suggested reading speed for German is 140-150 wpm; moreover, German also requires approximately 20% more space. Much thought has been given to assistance in translation tasks, yet the translator has to be proficient at handling subtitling software and observe compliance with all time and space constraints and gap rules at the same time. The only task that ceases to apply is colour-coding: this is neither needed to be produced from scratch in interlingual subtitling nor from a template in SDH translation as this should have been done for the SL already.

Two problems have, therefore, been identified: the first is that translators need to think about possible translations and edit them down accordingly, and the second is that translators have to adjust time-codes according to the rules of their TL. The first of these problems is relatively easy to solve if a transcript is available and the translators have an opportunity to use subtitling software of their own: feed the transcript into the CAT tool, translate it there using a TM and term base, take the result, feed it into the subtitling tool, and post-edit the product according to the TL wpm rates and line up the semantic units as needed. If the translator does not own any subtitling software or is required to translate in a cloud with no opportunity given to adjust time-codes according to language needs, another complication is added to the issue: Either subtitlers are limited in their ability to perform pure translation tasks or have to post-edit their own work and thus, carry out pure post-editing tasks, which takes a great deal of time.

This means I can map the following attributes to the needed function within the solution-oriented cycle:

| ATTRIBUTE | ISSUE | SOLUTION | FEATURE |
|---|---|---|---|
| **(a) dialogue available** | spotting and splitting is time-consuming | time-code, split according to line restrictions and minimum gap | Auto-cueing |
| **(b) repetitions & lyrics** | inconsistent translation or need to exit the software to consult previous translations | translate consistently, suggest whether or not lyrics need be translated | Auto-recognition of TM units of e.g. recurring mottos in a TV show, or lyrics with no need for translation |

|  |  |  |  |
|---|---|---|---|
| **(c) non-repetitive elements, but simple structure** | limited space allows only for concise language | automated suggestion of TM units and MT matches suitable for time and space constraints | TM & MT for suggestion to post-edit |
| **(d) customer specific or series specific terms and phrases** | need to exit the software to consult style guide | store in a database within the software | Term base for series/film specific terminology |
| **(e) Subtitling announcement (needed or not? → template by customer)** | need to exit the software to consult style guide | store rules in a customer-specific template within the software to consult | Customer template including reference files and formatting rules |
| **(f) long words that need replacement** | limited space allows only for concise language | omit, paraphrase, condense, summarise | Thesaurus to auto-recognise words and their length and auto-suggest synonyms or alternatives; also identify interjections, names, etc. |

*Table 3: Top-down analysis to map interlingual subtitle needs to possible CAT features*

These issues can be solved by integrating a tool that assists subtitlers in both their linguistic and technical work, such as a thesaurus for auto-suggestions of shorter alternatives and abbreviations auto-cueing or speech-to-text alignment of a transcript, term bases for show-specific terms, and perhaps most important in interlingual subtitling, a TM for repetitive but also non-repetitive elements with a simple sentence structure. What further attributes can be identified?

### 3.1.3. Common attributes of offline subtitles

This sub-section discusses attributes that apply to all forms of subtitling.

| ATTRIBUTE | ISSUE | SOLUTION | FEATURE |
|---|---|---|---|
| **Beginning and End Credit (colour, wording, duration, timing) → template** | need to exit the software to consult style guide | store rules in a customer-specific template within the software to consult | template |
| **Cue asterisks** | is it needed or not at the beginning of a subtitle file unclear → need to consult style guide | store rules in a customer-specific template within the software to consult | template |
| **How to handle speech that crosses shot changes** | there is no QA element to check if a subtitle runs over a shot change | store rules in a customer-specific template within the software to consult and have instances identified in a final check | template and QA element |
| **Semantic units** | after editing down, semantic units like article+noun in the same line are not observed and cannot be checked | identify semantic units in a final check | Final check to identify articles/prepositions and insert nonbreaking space or automatic line break after punctuation |

*Table 4: Top-down analysis to map general subtitling needs to possible CAT features*

Having discussed the above attributes in terms of their issues and possible solutions that can be translated into a function in a software program, I would like to summarise the functions to provide a better overview in the following spreadsheet and also to look into the availability of the key features mentioned in Sections 3.2 and 3.3.

| | ASR for auto-transcription | Auto-cueing | Music/intro/lyrics identifier | TM | Term base | Template | Thesaurus | Auto-colouring or tagging | OCR open subtitles | MT | QA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Intra** | X | X | X | X | X | X | X | X | X | X | X |
| **Inter** | | X | | X | X | X | X | | X | X | X |

*Table 5: Concise CAT requirements specific to subtitling forms*

Table 5 shows that, regardless of the subtitle form, an AVTEnT should have many features that not only help translators, but also subtitlers and master file creators. Since translators are the main target group of this thesis, I endeavour to investigate whether the features I named above are integrated in any subtitling tool and vice-versa. Therefore, Sections 3.2 et seqq. explore the availability of the mentioned features and allocate points to the subtitling or translation tool for each requirement fulfilled: the usefulness of the same features are field-tested in *4 Solution: Which CAT features should the AVTEnT have?* The points will be allocated as follows:

**0.5 points** – function available, but only conditionally (such as auto-cueing as a paid add-on)

**1 point** – function fully included in the application

## 3.2. Market analysis: Existing subtitling programs and their translation features

In this section, I discuss research conducted regarding which subtitling software programs have an integrated translation tool that is available both as freeware and commercially and whether this tool is useful. If there was a useful CAT tool or component integrated into subtitling software, I tested it and took inspiration to use and elaborate upon it in the proposal for a solution in Chapter 4.

If no solution is found, I would like to explore why and create a draft solution with some options and assumptions of the tool's usability that can be verified through a survey at a later point, which could then be implemented in a PhD dissertation. There could possibly be an easy way to fill the gap instead of designing a new feature or even software: perhaps a simple add-on for e.g. WinCAPS Qu4ntum (Screen Systems). It is thus vital to see what new developments have been achieved in the past 20 years.

After having received a clear ‘yes’ from academia and professional subtitlers to the question of whether a computer-aided translation tool in subtitling environments is useful at all, the next question of this thesis was examined: Which functionalities would the CAT tool need to have and what would be their benefit? How likely is a re-use of a translated subtitle where time and space constraints may not allow it? Consequently, I looked into what is already out on the market and, from this, compiled their practicable features for the introduction of a CAT tool for subtitling software.

After around 40 years of subtitling technology, there is a wide variety of subtitling software available on the market. These range from freeware, such as Aegisub or Subtitle Workshop for non-professional subtitling, as well as proprietary software, such as WinCAPS Qu4ntum by Screen Systems, FAB or EZTitles, through to cloud-based solutions. Therefore, an analysis of the existing software solutions will first be necessary. This analysis examines whether subtitling software has an integrated TM, term base, or thesaurus. If so, I then explore whether the software works with written templates only or with speech recognition as well. Which programs there are and what their features are, are explained in detail in this section. The question to be answered is not only whether programs are out there, but also if there are useful translation features to it. For a better overview, the following table is provided:

| **PRODUCT NAME** | **WEBSITE** | **FREEWARE?** | **TRANDLATION FEATURE** | **DESCRIPTION** |
|---|---|---|---|---|
| **Aegisub** | http://www.aegisub.org/ | Free software | No | |
| **Subtitle Workshop** | http://subworkshop.sourceforge.net/ | Free software for download | Yes | There is a side-by-side translation mode, but no functionality to export/import a TM, use a term base or aligner |
| **FAB** | https://www.fab-online.com/eng/subtitling/ | Licenced but no prices on website | No | |

| | | | | |
|---|---|---|---|---|
| **Subtitle Edit** | www.nikse.dk | Open source | No, but MT (Multi Translator) | MT for SV-DA subtitles with a database of 40k subtitles |
| **ONYX** | http://purpledragonmedia.co.uk/onyx-subtitler/ | Free 14-day trial license | Yes | No TM, no import or export of TUs |
| **Sfera Studios** | https://www.sferastudios.com/ | Agency-only license | Yes, but only for such providers as Netflix | Cloud-based translation tool for subtitles for in-house translator of, for example, Netflix |
| **WADOS** | http://www.webtrans.com.tw/service-e.asp | Commercially not available | Yes, but only in-house | Translation platform exclusively-developed by Webtrans, a Taiwanese translation service |
| **FilmTit** | http://filmtit.namesny.com/ | Open source online tool | yes | TM for CS-EN subtitles with MT function |

*Table 6: Market analysis of existing subtitling programs with and without CAT tool elements*

As explained in *1.1.1 Translation memory (TM)*, the XLIFF and TMX file standards enable an easier exchange of translation memory files. In order for subtitle files to be exchangeable, the file should have the extension *.STL 'to indicate that the datafile conforms to the EBU standard subtitle data exchange format' (EBU 1991: 3).[14] This format is only used for a quick exchange, but there are many other file formats that can be created by subtitling software. However, not all subtitle formats are equally supported by all subtitling programs mentioned above.

MUSA (MUltilingual Subtitling of multimediA content, 2002-2004) was a research project that examined the feasibility of statistical machine translation in subtitling. The project's aim was to create a multimedia system that generated speech-to-text subtitles, analysed the linguistic structure and then auto-translated them into other languages by 'combining a Machine Translation engine with a Translation Memory and a Term Substitution module' (MUSA, http://sifnos.ilsp.gr/musa). The project failed as no substantial

---

[14] Subtitle Data Exchange format: https://tech.ebu.ch/docs/tech/tech3264.pdf

results could be achieved. Why did they fail? Díaz Cintas (2015) concluded that the reason for the failure was that speech recognition technology was not as developed as it is today; furthermore, the volume of necessary subtitle data of professional quality was low at the time.

This is something that SUMAT (SUbtitling by MAchine Translation) had compensated for, which was mentioned earlier in *2.2 Lower costs, higher productivity*. A group of subtitling companies from seven EU member states collaborated with technical partners on a project funded by the European Commission under the subject *Information and Communication Technology* (ICT) *applications* of the *Competitiveness and Innovation Framework Programme* (CIP), whose objective was to make 'Europe the most competitive and dynamic knowledge-based economy in the world' (CORDIS 2014). Similar to the CAT tools' evolution when researchers collected data to set up TMs in the early 1980s that were meant to assist in the translation process (see Section 1.1), the SUMAT participants collected subtitle files in seven language pairs with some consisting of up to two million subtitles to create a corpus. This corpus was then used to train statistical machine translation (SMT) systems (Etchegoyhen et al. 2013) and develop MT with high-quality outputs in subtitling. Other projects in the domain of linguistic research data for the training of SMT are, for example, the 'sentence-parallel Czech-English corpus compiled at the Institute of Formal and Applied Linguistics (ÚFAL): freely available for non-commercial research purposes' (ÚFAL online) and Flanagan's *Example-Based Machine Translation to translate DVD Subtitles* (2009). These projects show that research into the increase of productivity in subtitling has moved away from merely tackling technical issues to proceed instead with linguistic aspects of the profession.

Has this trend, however, arrived in the industry of subtitling software developing companies and amateur developers? I have picked out three of the programs named in the table above that have an integrated translation tool to test its functionality: the free software Subtitle Workshop version 6.0 available for download, the proprietary software WinCAPS Qu4ntum by Screen Systems, and FilmTit for which a TM was created by a group of students in a Bachelor's programme project and then taken over and developed further by one person on Github.

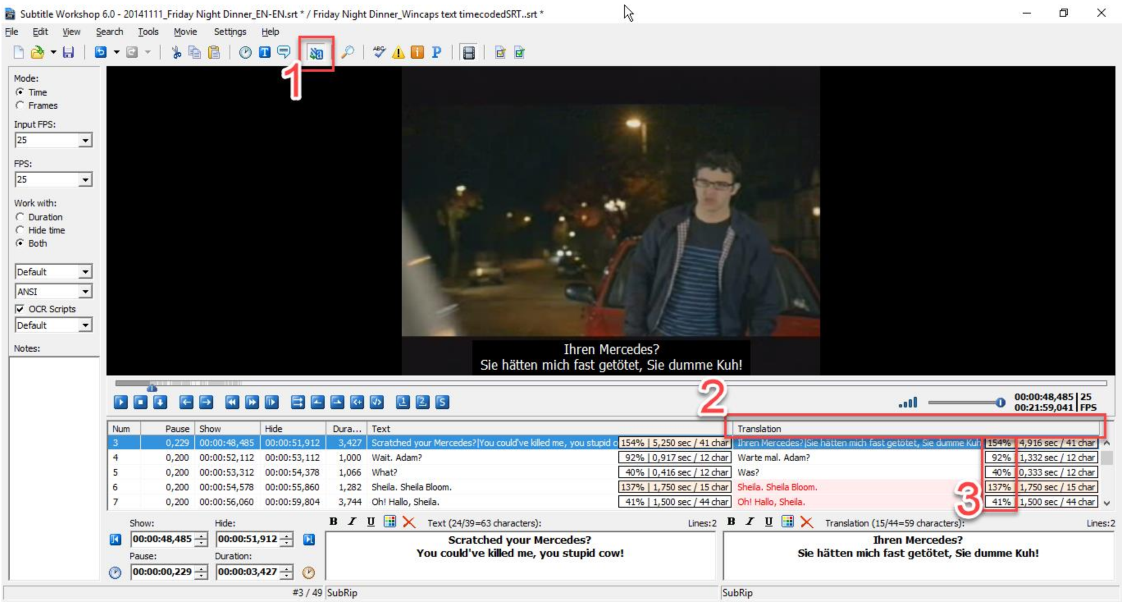
### 3.2.1. Subtitle Workshop

**What is it?** – Subtitle Workshop (https://sourceforge.net/projects/subworkshop/) is an application available online for download to create and translate subtitles. This free software has an integrated translation function where you see the translated subtitle immediately projected onto the video, but it does not store translation units to a TM or allow for a TM export/import.

**Translation possible?** – For **interlingual** translation, the translator only needs to turn on *translator mode* by clicking on the button (see #1 in Figure 6 below) or clicking *Edit/Translation* and selecting *Translator Mode* or pressing *Ctrl+U*. Below the video screen, a translation column appears side-by-side with the original subtitles (see #2). To select which subtitles (the original or the translation) should be projected onto the video, the translator needs to swap the two columns by clicking *Edit/Translation/Swap*. Another way would have been to click *Movie/Display/Translation*, but the first option is quicker to find. It also takes time to find the reading speed setting, which can be set via *Tools/Information and errors/Settings (Alt+I)*; this path is very unintuitive as there is an item called *Settings* in the menu bar, which led to a completely different setting area than the one above.

I selected the tab *Advanced*, where I could set the *CpS* from the preset of 20 to a value of 13 after having translated the first subtitles. The application now shows that my proposed subtitles exceed the maximum reading speed in percentage (see #3). As a way to build a TM, I suggest to *Save Translated As* and *Save Original As* a SubRip file and align the documents with a CAT tool or aligner that supports files with an *.srt extension.

For an **intralingual** translation with no transcript available, however, there is no option to work from a template due to the lack of ASR and auto-transcription with post-editing options. The creation of subtitles has to be done from scratch with no technological assistance.



Subtitle Workshop 6.0 - 20141111_Friday Night Dinner_EN-EN.srt * / Friday Night Dinner_Wincaps text timecodedSRT..srt *
File
Edit
View
Search
Tools
Movie
Settings
Help
1
2
3
Ihren Mercedes?
Sie hätten mich fast getötet, Sie dumme Kuh!

Mode:
○ Time
○ Frames
Input FPS: 25
FPS: 25
Work with:
○ Duration
○ Hide time
● Both
Default
ANSI
☑ OCR Scripts
Default
Notes:

00:00:48,485 | 25
00:21:59,041 | FPS

| Num | Pause | Show | Hide | Dura... | Text | | Translation | | |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 0,229 | 00:00:48,485 | 00:00:51,912 | 3,427 | Scratched your Mercedes?\|You could've killed me, you stupid c | 154% \| 5,250 sec / 41 char | Ihren Mercedes?\|Sie hätten mich fast getötet, Sie dumme Kuh | 154% | 4,916 sec / 41 char |
| 4 | 0,200 | 00:00:52,112 | 00:00:53,112 | 1,000 | Wait. Adam? | 92% \| 0,917 sec / 12 char | Warte mal. Adam? | 92% | 1,332 sec / 12 char |
| 5 | 0,200 | 00:00:53,312 | 00:00:54,378 | 1,066 | What? | 40% \| 0,416 sec / 12 char | Was? | 40% | 0,333 sec / 12 char |
| 6 | 0,200 | 00:00:54,578 | 00:00:55,860 | 1,282 | Sheila. Sheila Bloom. | 137% \| 1,750 sec / 15 char | Sheila. Sheila Bloom. | 137% | 1,750 sec / 15 char |
| 7 | 0,200 | 00:00:56,060 | 00:00:59,804 | 3,744 | Oh! Hallo, Sheila. | 41% \| 1,500 sec / 44 char | Oh! Hallo, Sheila. | 41% | 1,500 sec / 44 char |

Show: 00:00:48,485
Hide: 00:00:51,912
Pause: 00:00:00,229
Duration: 00:00:03,427

Text (24/39=63 characters): Lines:2

Scratched your Mercedes?
You could've killed me, you stupid cow!

Translation (15/44=59 characters): Lines:2

Ihren Mercedes?
Sie hätten mich fast getötet, Sie dumme Kuh!

#3 / 49 SubRip
SubRip

*Figure 6: UI of Subtitle Workshop with translation mode activated (source: Channel 4’s Friday Night Dinner*

**Quality levels** – There are many settings that are not visible by default and are easily adjustable to the user's demand. A quick quality check using WinCAPS shows that with the default setting of *Backward and Forward time* of 0.5 second and the default reading speed of 20 cps, the subtitles mostly run over shot changes and, with the most frequent rate of 180 wpm, exceed the recommended reading speed for German-language subtitles of 140-150 wpm. For professional time-coding, including moving forwards and backwards in the video frame by frame rather than the default setting of 0.5 second, I needed to click *Settings/Settings/Video preview* and enter 0.040 seconds in the field *Backward and Forward time* for a 25 FPS video as a conversion rate for frames to seconds: yet another unintuitive place for this setting.

**Useful?** – For a quick and dirty hack, Subtitle Workshop is a nice application for non-professional subtitling and fansubbing but, unfortunately, it is not useful for the purpose of this thesis. Overall, the settings are unintuitive and it takes some time to find out where they are placed. The question of this sub-section; namely, if there is a useful translation feature to the application, can be answered with a simple 'no'. Yet, as always with open source applications, anyone can modify and improve it under the terms of the GNU General Public License. If, for an analytical point of view, I also look at the possible solutions that I identified in *3.1 Forms of subtitling and their technological needs*, I can say that the components that are covered by the software receive 0 points—not even a linguistic check is included—since the software itself is only available for English, Bulgarian, and Russian (Table 7).

| | **ASR for auto-transcription** | **Auto-cueing** | **Music/intro/ lyrics identifier** | **TM** | **Term base** | **Template** | **Thesaurus** | **Auto-colouring or tagging** | **OCR open subtitles** | **MT** | **QA** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Intra** | X | X | X | X | X | X | X | X | X | X | X |
| **Inter** | | X | | X | X | X | X | | X | X | X |
| **SW** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Table 7: CAT requirements fulfilled by Subtitle Workshop 6.0*

As the table shows, useful features for both intralingual and interlingual translation do not exist in the application.

**Note**: The latest version available on Sourceforge is 6.0 whereas URU Works, the initial builder of the software (then-called URU Soft), has released a version 6.01 on their website http://www.uruworks.net/download.html. This software not only has a different name, Subtitle Workshop XE, but also a different layout with a completely different user interface and a more user-friendly manual[15] online. However, apart from the audio waveform extractor and the option to import scene change times, the core functions remain the same despite the new layout. Additionally, there is a lower quality in usability, such as the inability to jump back and forward in the video, which means that back and forward is possible in the timeline, but not in the video without having to press the *Play* button.

### 3.2.2. WinCAPS

**What is it?** – WinCAPS Qu4ntum, now called Wincaps Q4, is subtitle creation software for professional offline subtitles, and a product of Screen Subtitling Systems Ltd. The company was founded in 1976 and pioneered in the field of subtitling technology. Their products range from pay-as-you-go and monthly or yearly plans to full proprietary licences for both offline and live subtitle software for freelancers, in-house subtitlers, and localisation vendors: the software works with the help of a USB dongle and a licence code. The standard features of the software are as follows: move video frame-by-frame, automated detection of shot changes and speech presence, edit/fix in and out times, customer templates, reading speed and duration at a glance, and a side-by-side translation window if needed. With an optional upgrade to Q4 Pro, the user can also auto-sync time-codes with a transcript (*Automatic Timing Alignment*), use the integrated function for the ASR software Dragon® NaturallySpeaking,[16] automate extraction of dialogue from a transcript (*Script Extractor*), and automatically colour-code speakers (see Screen Subtitling Systems).

**Translation possible?** – There is, in fact, a translation function, which can be activated by the commands *File/Translation/Open* or *…/New*. A side-by-side translation view opens where the original, if there is a transcript, stays in the left column, while the translation can be entered in the right column. The translated subtitles appear automatically in the preview window on the video. In addition to the above mentioned standard features, the software allows for the creation of many different customer templates in regards to font, positioning, minimum and maximum duration, and minimum gap. The software observes these configurations

---

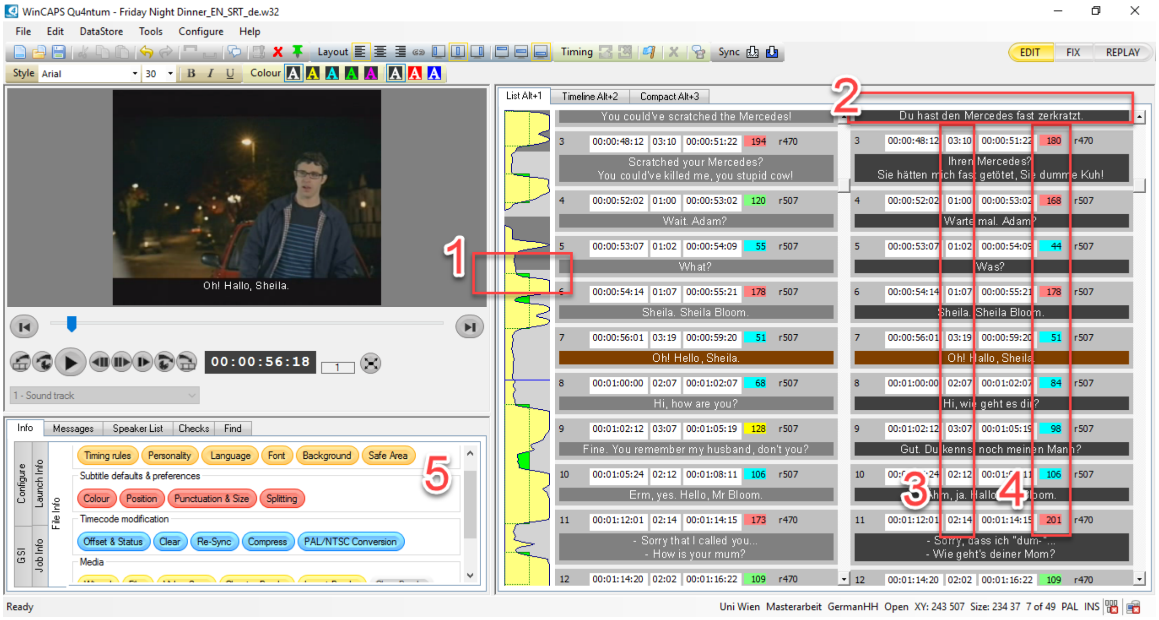
[15] Manual downloadable at http://www.urusoft.net/downloads/docs/sw/Getting%20started.pdf (last accessed 2017-07-05)

[16] See more on https://www.nuance.com/dragon/dragon-for-pc/home-edition.html

automatically; for example, when you press enter to add a subtitle, the minimum gap between the last and the present subtitle and the minimum duration for the new subtitle is set for you, and when you start to type, the cursor is positioned as configured. Additionally helpful is the use of colours for reading speed. When the wpm rate of a subtitle is satisfactory, the field containing the individual rate stays green; once it is close to the limit, however, it turns yellow and, when it exceeds the maximum wpm rate, it turns red. The subtitling process can be easily navigated via NUM keys with numbers being allocated to commands; for instance, *play* is 5, *step forward* is 6, *step back* is 4 and so on.

When you open the translation view and start translating, the translation is saved automatically in a new file with the respective language extension to the same folder as the source file. Wincaps' native file type is w32, so all files are stored with the extension *.w32. In the case of my translation from UK English to DE German, the extension is *_de.w32. The next time I open either file, source, or translation, the file opens in monolingual view and I must add the translation file once more. For improved usability, the file could be stored in a project folder with the option to choose from a range of files that can be opened: monolingual SL file, monolingual TL file, or a merge to a bilingual product.

A feature that comes in very handy is the automated shot change detection. The program creates helper files for each video and, with the aid of these, it detects speech presence, audio levels, and indices. If these helper files are not in the folder of the subtitle file, you can easily create them as follows: in the toolbox (#5 in figure below), click on the tab *Info,* then on the tab *File Info*, and select in row *Media* the *Wizard* button. The wizard screens the chosen video and identifies and marks shot changes and speech audio waves. These shot changes, however, only help in navigating through the video while creating the subtitles and are not fixed by the program itself; thus, the user must carry out the task of fixing out times before a shot change manually. A detected shot change can be seen in figure below (#1). The translation column is on the right (#2), with duration marked in #3 and wpm rates in #4.



| | | | | | |
|---|---|---|---|---|---|
| | | | | | You could've scratched the Mercedes! |
| 3 | 00:00:48:12 | 03:10 | 00:00:51:22 | 194 | r470 |
| | | | | | Scratched your Mercedes?<br>You could've killed me, you stupid cow! |
| 4 | 00:00:52:02 | 01:00 | 00:00:53:02 | 120 | r507 |
| | | | | | Wait. Adam? |
| 5 | 00:00:53:07 | 01:02 | 00:00:54:09 | 55 | r507 |
| | | | | | What? |
| 6 | 00:00:54:14 | 01:07 | 00:00:55:21 | 178 | r507 |
| | | | | | Sheila. Sheila Bloom. |
| 7 | 00:00:56:01 | 03:19 | 00:00:59:20 | 51 | r507 |
| | | | | | Oh! Hello, Sheila. |
| 8 | 00:01:00:00 | 02:07 | 00:01:02:07 | 68 | r507 |
| | | | | | Hi, how are you? |
| 9 | 00:01:02:12 | 03:07 | 00:01:05:19 | 128 | r507 |
| | | | | | Fine. You remember my husband, don't you? |
| 10 | 00:01:05:24 | 02:12 | 00:01:08:11 | 106 | r507 |
| | | | | | Erm, yes. Hello, Mr Bloom. |
| 11 | 00:01:12:01 | 02:14 | 00:01:14:15 | 173 | r470 |
| | | | | | - Sorry that I called you...<br>- How is your mum? |
| 12 | 00:01:14:20 | 02:02 | 00:01:16:22 | 109 | r470 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | Du hast den Mercedes fast zerkratzt. |
| 3 | 00:00:48:12 | 03:10 | 00:00:51:22 | 180 | r470 |
| | | | | | Ihren Mercedes?<br>Sie hätten mich fast getötet, Sie dumme Kuh! |
| 4 | 00:00:52:02 | 01:00 | 00:00:53:02 | 168 | r507 |
| | | | | | Warte mal. Adam? |
| 5 | 00:00:53:07 | 01:02 | 00:00:54:09 | 44 | r507 |
| | | | | | Was? |
| 6 | 00:00:54:14 | 01:07 | 00:00:55:21 | 178 | r507 |
| | | | | | Sheila. Sheila Bloom. |
| 7 | 00:00:56:01 | 03:19 | 00:00:59:20 | 51 | r507 |
| | | | | | Oh! Hallo, Sheila. |
| 8 | 00:01:00:00 | 02:07 | 00:01:02:07 | 84 | r507 |
| | | | | | Hi, wie geht es dir? |
| 9 | 00:01:02:12 | 03:07 | 00:01:05:19 | 98 | r507 |
| | | | | | Gut. Du kennst noch meinen Mann? |
| 10 | [illegible] | 02:12 | [illegible] | 106 | r507 |
| | | | | | [illegible]hm, ja. Hallo, [illegible]loom. |
| 11 | 00:01:12:01 | 02:14 | 00:01:14:15 | 201 | r470 |
| | | | | | - Sorry, dass ich "dum-"...<br>- Wie geht's deiner Mom? |
| 12 | 00:01:14:20 | 02:02 | 00:01:16:22 | 109 | r470 |

*Figure 7: UI of WinCAPS Qu4ntum with translation mode activated (source: Channel 4's* Friday Night Dinner *S01E05, 2011–)*

**Quality levels** – The user can navigate with precision by moving back and forward frame-by-frame in a simultaneous track of both video and audio. Thanks to this feature, the user can observe shot changes and speech start/end easily. A quality check at the end, however, revealed another failure: an excess in minimum and maximum duration, line length, reading speed, number of maximum rows, empty lines and leading/trailing/multiple spaces, and, of course, spelling. The program can handle, create, edit, and export a range of file types, and is therefore, a powerful tool for subtitling. Nevertheless, the translation side still has room for improvement.

**Useful?** – No matter how powerful the tool is in terms of subtitling, it still lacks translation features that deal with language matters, apart from a spelling checker. Although translation is possible, an export of translation units is not included: this is the same issue as for Subtitle Workshop. Here, again, comes the workaround of exporting the EN and DE files into an SRT, feeding them into a CAT tool to align them, and saving the TM. Even so, information, such as wpm, line length, limited number of lines and characters per line, cannot be stored in the TM and the next translation has to be done either in a separate CAT tool where the TM can be used or in the subtitling program where it cannot be used, but the video can be watched and spotted. An idea would be to have a TM where you can preset the possible limitations and have the TM show results with a lower match percentage if the attributes do not suit the settings. This could then be used within the application without the need to switch screens.

More on this follows in Chapter 4. Table 8 shows which requirements were fulfilled with an explanation below.

| | ASR for auto-transcription | Auto-cueing | Music/intro/ lyrics identifier | TM | Term base | Template | Thesaurus | Auto-colouring or tagging | OCR open subtitles | MT | QA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Intra** | X | X | X | X | X | X | X | X | X | X | X |
| **Inter** | | X | | X | X | X | X | | X | X | X |
| **Q4** | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0.5 | 0 | 0 | 1 |

*Table 8: CAT requirements fulfilled by WinCAPS Qu4ntum*

Wincaps offers ASR for transcription as an option in the higher-priced Q4 Pro product, for which it scores 0.5 points, but the ASR is speaker-dependent and therefore not applicable to auto-transcription, which is why it also receives only another 0.5 points. This speaker-dependent dictation and speech recognition software is a Dragon® product called NaturallySpeaking and can be used with the standard Q4 version as well if the user has a separate licence for the software; however, it takes more resources to process and only work through an in-built workaround (see *4.1.4 Field test (2): WinCAPS and Dragon® NaturallySpeaking)*.

As explained above in the *What is it* paragraph, the option Q4 Pro version also includes an auto-cueing feature called *Automatic Timing Alignment*. This feature only works if a transcript is available, which usually is not provided for (for instance) SDH. If, however, the transcript is available, such a tool can be of tremendous help for subtitling. As a possible workaround, the dialogue could be transcribed via ASR into a document, which then could be fed into the subtitling tool with the automatic timing alignment. Yet again, this is just a workaround and therefore takes more time than simply creating the subtitles on the go. This add-on feature, thus, receives 1 point in total: 0.5 points for being useful in interlingual (but not intralingual) subtitling and another 0.5 points for the inclusion of the auto-cueing feature that is, however, only applicable with a separate, proprietary licence of the Dragon® software and is a time-consuming workaround.

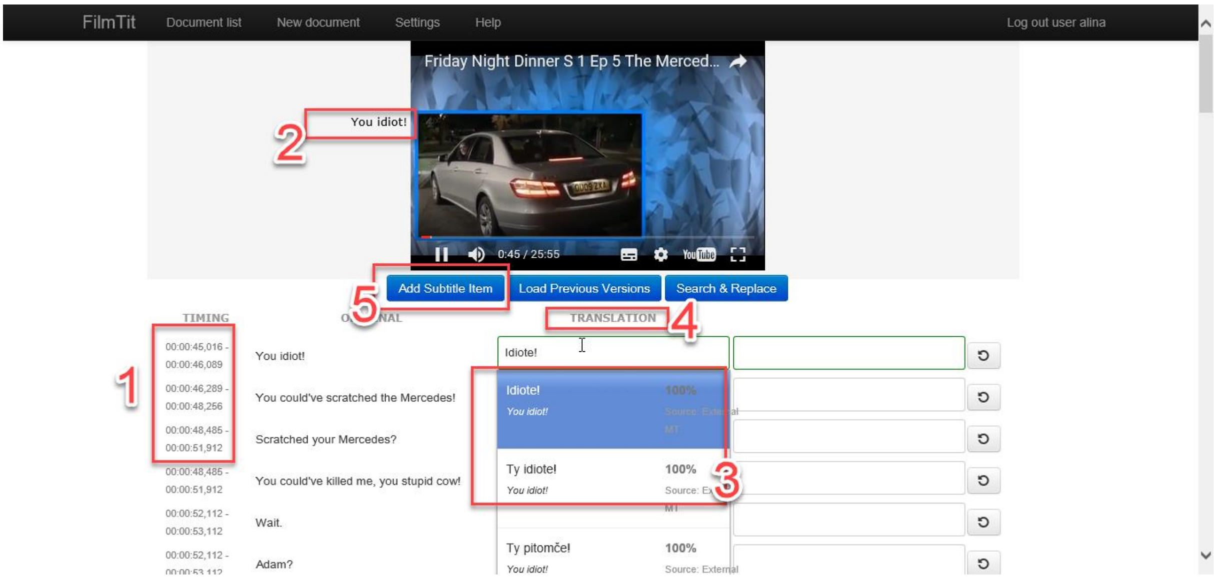
Another point goes to the software for having an integrated feature to create a customer template, which is expandable; hence, it should be easy to add more template features to it. For the auto-colouring feature, it scores 0.5 points as it is included in the standard version although it can only be used with a script. There is an integrated function for spelling checks, which is certainly expandable and therefore gets 1 point.

The resulting 4.5 points out of 11 possible solutions, when compared to Subtitle Workshop, is a good result with high potential for advancement. The features that can be used and evolved are discussed further in Chapter 4.

### 3.2.3. FilmTit

**What is it?** – FilmTit is a Bachelor’s degree programme project led by six students from the Faculty of Mathematics and Physics of Charles University in Prague, who created an online application for fan-based non-professional subtitle translation (fansubbing) with a built-in TM and MT for English-Czech and Czech-English translations. This application was taken over and further developed on Github by Matúš Námešný.

**Translation possible?** – Users can easily register and upload local files that are not going to be stored throughout sessions, with the consequence that every time the application is started, the user needs to upload the file or a URL. A file upload of WAV, AVI, or MP4 did not work, so I looked up the source on YouTube and entered the URL into the application. This function embeds the video into the application and jumps to the respective position according to the set time (#1 in Figure 8) in the video when the user clicks in the translation field. On the left-hand side of the video, the current SL subtitles appear (#2). The translation field offers MT segments (#3) for translation into Czech in the central column (#4), and a post-editing field in the right column. Once the file is uploaded, there is the possibility to adjust one singular setting, which is the *Maximum number of suggestions to show for each line*. When the translation is finished, the user has the following options to convert his or her subtitles into a TXT or SRT file: Translated version (untranslated parts left out), Translated with original where untranslated, Source, Post-edited version (non-post-edited parts left out), Post-edited with translated where unpost-edited. Again, no TM export (or import) is possible.



You idiot!

| TIMING | O NAL | TRANSLATION |
|---|---|---|
| 00:00:45,016 - 00:00:46,089 | You idiot! | Idiote! |
| 00:00:46,289 - 00:00:48,256 | You could've scratched the Mercedes! | |
| 00:00:48,485 - 00:00:51,912 | Scratched your Mercedes? | |
| 00:00:48,485 - 00:00:51,912 | You could've killed me, you stupid cow! | |
| 00:00:52,112 - 00:00:53,112 | Wait. | |
| 00:00:52,112 - 00:00:53,112 | Adam? | |

Idiote! 100%
*You idiot!* Source: External
MT

Ty idiote! 100%
*You idiot!* Source: Ex
MT

Ty pitomče! 100%
*You idiot!* Source: External

*Figure 8: UI of FilmTit (source: Channel 4's* Friday Night Dinner *S01E05, 2011–)*

**Quality levels** – The use of the application is intuitive at first glance but also buggy, so the videos could not be uploaded; furthermore, the documents were created five times with each of the documents being non-editable due to some errors. The help section contains screenshots and explanations of each function and section of the application, but does not contain the format requirements and no email address to contact in case of difficulties with the application. The settings are limited to only one attribute, so settings concerning reading speed, line length, maximum number of characters per line, etc. are not available. The quality of the MT segments could not be evaluated since I do not work with the Czech language. Particularly unhelpful was that, although there is a source mentioned in the MT segment (see #3) and it says *Source: External*, it does not state where it comes from; for instance, genre, show title, or context. These attributes show that this application is clearly designed solely for interlingual subtitling purposes.

**Useful?** – Time-codes (#1 above) appear hard-coded in the application and are not adjustable, but the user can *Add a Subtitle Item* (#5) for which she or he needs to know the exact time as there is no spotting function included in the application. There is no way to import and use one's own TM or to export one in any TM export file type. The only way to receive your translations is to export them as an SRT or TXT file and, as with SW and Q4 above, feed them into your CAT tool and align them. As I can see, an alignment tool is essential for translators in subtitling. Yet, due to the lack of an import function, the TM is not applicable for the tools at hand. In regards to my possible solution, we can now examine the elements that are useful by referring to Table 9:

| | ASR for auto-transcription | Auto-cueing | Music/intro/lyrics identifier | TM | Term base | Template | Thesaurus | Auto-colouring or tagging | OCR open subtitles | MT | QA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Intra** | X | X | X | X | X | X | X | X | X | X | X |
| **Inter** | | X | | X | X | X | X | | X | X | X |
| **FT** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

*Table 9: CAT requirements fulfilled by FilmTit*

Concerning intralingual subtitling, there is no given function. To start subtitling, the user is required to upload an SRT file containing subtitles and time-codes in order to translate from there. Since there is no ASR, a transcript cannot be created and, due to the lack of a cueing function, no further subtitles (for example, for sound description) can be added. I was hoping for a TM as this was suggested in the documentation of the bachelor's project, but this was not included. The only function that the online application features is the MT function for EN-CS and CS-EN.

## 3.3. Market analysis: Existing CAT tools and their use in subtitling

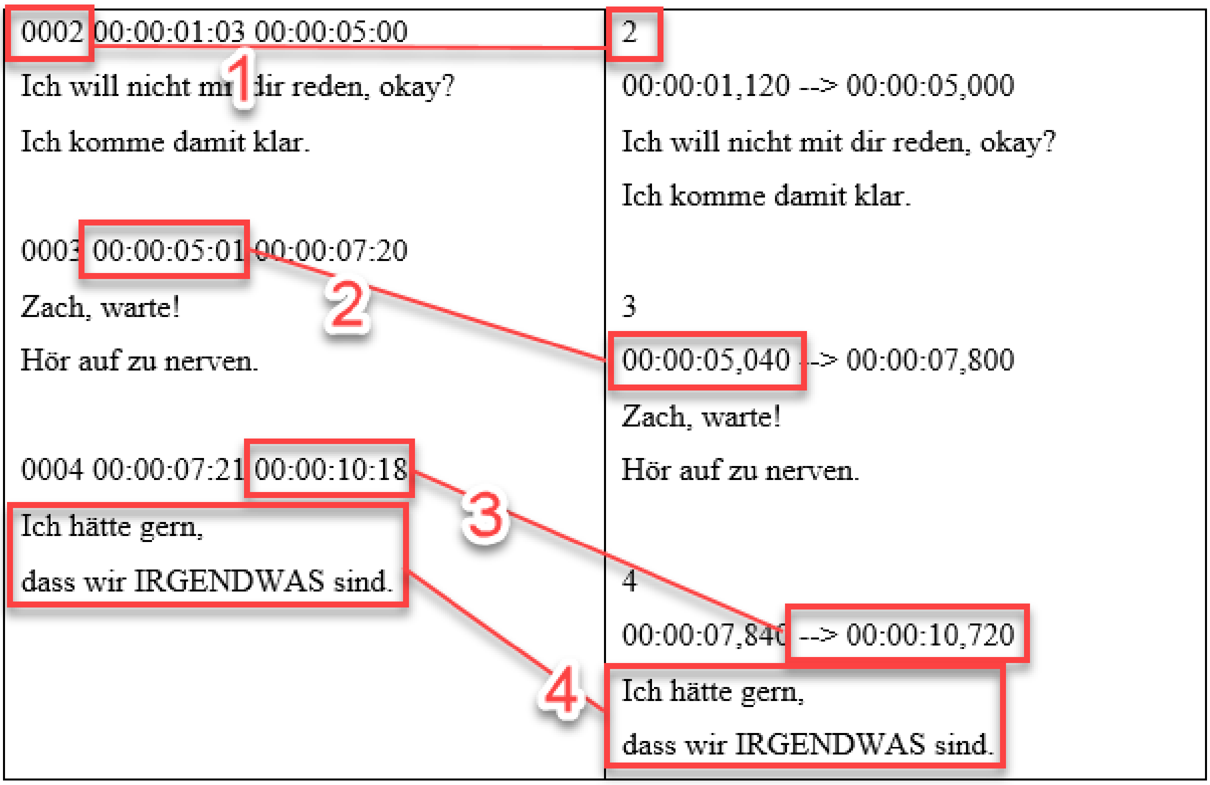
Having discovered that much has been researched but effectively little achieved in the field of subtitling tools with integrated CAT tool functions, the next thing to do was to explore the opposite: whether there were any solutions in CAT tools that are useful for subtitling, including translating an SRT file, watching a video, using TM and MT during the translation process, and quality assurance elements for subtitle line length, to name one example. The only three CAT tools with integrated subtitling features that I identified are explained in this section.

There are several CAT tools on the market, and most of them can handle the common file types, such as PDF, regular MS Word and Excel files, exchange files from other applications such as XLIFF and XML, or HTML content. SRT and other subtitle file formats, however, are usually not among the supported file types. Only the CAT tools Transit NXT, SDL Trados Studio, and Memsource offer partial solutions to my question. Since the analysis has shown that there are only a few such tools available, this section aims to find solutions by experimenting, documenting, and analysing the subject of which CAT tool features are useful for an AVTEnT. I begin by trying the two most established CAT tools in the market, which actually laid the foundation for CAT tools: STAR Group’s Transit NXT and SDL Trados Studio 2014.

### 3.3.1. SDL Trados Studio 2014

**What is it?** – SDL Trados is a CAT tool developing and service-providing company that sells licences to freelance translators as well as to LSPs (Language Service Providers). Their solutions range from TM through TMS and MT to software localization and translation management tools. In 1984, Hummel and Knyphausen founded Trados (TRAnslation & DOcumentation Software GmbH) to serve as an LSP to IBM. They developed the terminology management tool *MultiTerm* in 1990 and the first *Translator’s Workbench* in 1994. In 2005, Trados was acquired and integrated by SDL. With their translation service tools, they were the forerunners of computer-assisted translation with term bases that included fuzzy matches for incorrectly spelt words, an interface that looked like a dictionary, an alignment tool created by Matthias Heyn, and the software localization tool Passolo (see SDL Trados n.d.: online). SDL Trados can handle a range of file formats from Microsoft Word and Excel through PDFs and Adobe InDesign files to XML and HTML files. Up until the version SDL Trados 2015, SubRip files were not supported.

**Subtitle file translation possible?** – Since SDL Trados Studio 2014 did not initially support SubRip files, I decided to translate from an RTF as this is a supported file type by SDL. Subsequently, I exported my subtitle file from WinCAPS as an RTF and fed it into the CAT tool. Figure 9 below shows what a subtitle file’s formatting looks like when exported as an RTF (Rich Text Format, on the left in figure below) as compared to an SRT (on the right):



```
0002 00:00:01:03 00:00:05:00
Ich will nicht mi dir reden, okay?
Ich komme damit klar.

0003 00:00:05:01 00:00:07:20
Zach, warte!
Hör auf zu nerven.

0004 00:00:07:21 00:00:10:18
Ich hätte gern,
dass wir IRGENDWAS sind.
```

```
2
00:00:01,120 --> 00:00:05,000
Ich will nicht mit dir reden, okay?
Ich komme damit klar.

3
00:00:05,040 --> 00:00:07,800
Zach, warte!
Hör auf zu nerven.

4
00:00:07,840 --> 00:00:10,720
Ich hätte gern,
dass wir IRGENDWAS sind.
```

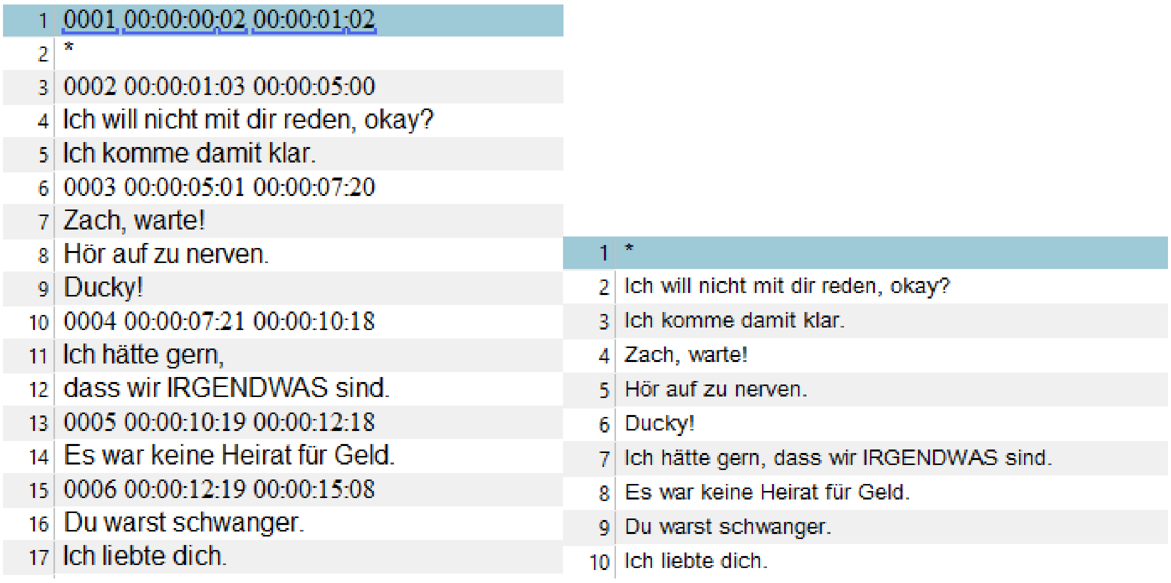
*Figure 9: Examples of a subtitle file exported as an RTF vs. SRT*

#1 shows that the RTF's formatting for the **subtitle number** and in and out times appear in a row, while the SRT's formatting is split into two lines. The subtitle number has four digits in the RTF file and only the respective number with no zeros in the SRT file. #2 shows that the **in time and out times** in the RTF are expressed in seconds and frames while the those in the SRT file appear converted into seconds and milliseconds with the conversion rate of 0.40 seconds per frame and using a comma as a delimiter. #3 shows that the **out times** in the SRT are separated by an arrow from the in times. #4 shows that the **subtitle's line breaks** are formatted in the same way for both file formats. Depending on the file format, the formatting of subtitle numbers and in and out times is different, thus the following coding can be identified:

| The formatting for **RTF** is: | The formatting for **SRT** is: |
|---|---|
| dddd hh:mm:ss:ff hh:mm:ss:ff | ddd |
| subtitle's first line | hh:mm:ss,ms --> hh:mm:ss,ms |
| subtitle's second line | subtitle's first line |
| | subtitle's second line |

When I fed the RTF into SDL Trados Studio 2014, the time-coded transcript was segmented by lines. Each line break constituted its own segment, and each time-code was displayed as its own segment; thus, all time-codes appeared as translatable segments (see Figure 10), which would not help translators as they would then have to ensure that they do not make any changes to the time-codes. If they did so, without having the possibility to properly spot them, this could corrupt the file. A way to watch the video within the application and to adjust the time-codes accordingly would therefore be useful.

The SDL Community Developers made a free app called ‘SubRip File Type’, which is available in the SDL AppStore (appstore.sdl.com) for signed up users. This app creates an additional file type within the settings of SDL Trados Studio 2014, which makes it possible to open SubRip files for translation as well as for alignment. In the later Studio version 2015 SR2, this file type definition is available by default, so an app installation is no longer needed. For the analysis in this thesis, however, I had to install it. After installation, the app integrated into the tool so that the segments containing time-codes were hidden from translation. To set the limitation of characters per line, for example to 38, the user only needs to adjust the file type definition (click *File/Options/File Types/SubRip/Common* and enter *38*). The figure below shows an RTF segmentation (left) using the default RTF file type definition and an SRT segmentation (right) using the SubRip File Type definition.



1 0001 00:00:00:02 00:00:01:02
2 *
3 0002 00:00:01:03 00:00:05:00
4 Ich will nicht mit dir reden, okay?
5 Ich komme damit klar.
6 0003 00:00:05:01 00:00:07:20
7 Zach, warte!
8 Hör auf zu nerven.
9 Ducky!
10 0004 00:00:07:21 00:00:10:18
11 Ich hätte gern,
12 dass wir IRGENDWAS sind.
13 0005 00:00:10:19 00:00:12:18
14 Es war keine Heirat für Geld.
15 0006 00:00:12:19 00:00:15:08
16 Du warst schwanger.
17 Ich liebte dich.

1 *
2 Ich will nicht mit dir reden, okay?
3 Ich komme damit klar.
4 Zach, warte!
5 Hör auf zu nerven.
6 Ducky!
7 Ich hätte gern, dass wir IRGENDWAS sind.
8 Es war keine Heirat für Geld.
9 Du warst schwanger.
10 Ich liebte dich.

*Figure 10: Screenshot of an RTF and an SRT in the SDL Trados Studio 2014 interface (source:* OC California *S02E05 (2003–2007) dubbed into German (USA: 2004-12-09, GER/AUT: 2005-08-03)*

To avoid the risk of changing the time-codes and, at the same time, to make intelligent segmentation possible, I recommend using an SRT file as a source in combination with the *SubRip File Type* definition. Once the translation is finished, a file including all source time-codes can be exported as either a bilingual or monolingual document. If the user is knowledgeable about coding, she or he could define a file type with parsing rules that would command the program to hide the subtitle numbers and time-codes.

**Quality levels –** The user can translate a file with the help of a TM, but not spot it and so, not be able to adjust time-codes according to the TL. Using a term base is possible which raises consistency. There is a range of quality checks included in the tool, where limitation to a certain number of characters per line is also possible; therefore, the translator is able to check compliance with style guide rules for each segment. However, since the program segments the file by sentence rather than subtitle, the translator does not know if the sentence for translation functions as part of one subtitle or if it constitutes an individual subtitle. Additionally, since there is no way to watch the video while translating in the same application, context is missing for the translator and decision-making for or against a translation may be restricted as a result.

**Useful?** – The user can make use of a TM and MT as a paid add-on from the SDL Language Cloud, term bases, and other useful plugins, such as a thesaurus and web search, which are already available in the AppStore. These include (among others) MT AutoSuggest, Web Lookup!, and IntelliWeb Search. The useful parts of a translation tool, the TM, term base, and the optional MT, are not available in a subtitling tool whereas the useful parts of a subtitling tool (video, adjusting time-codes and reading speed) are not available in SDL Trados. The company, however, gives anyone who is good at coding the opportunity to build a plugin and make it available to searchers on their SDL AppStore (appstore.sdl.com). With a huge database of apps, there is a good chance that, someday in the future, an add-on for the following functions might become available: play a video and go backwards and forwards frame-by-frame, adjust time-codes accordingly, and set reading speeds to the respective TL requirements. The following table shows the previously identified CAT tool features that are relevant in subtitling:

| | ASR for auto-transcription | Auto-cueing | Music/intro/lyrics identifier | TM | Term base | Template | Thesaurus | Auto-colouring or tagging | OCR open subtitles | MT | QA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Intra** | X | X | X | X | X | X | X | X | X | X | X |
| **Inter** | | X | | X | X | X | X | | X | X | X |
| **SDL** | 0 | 0 | 0 | 1 | 1 | 0.5 | 0.5 | 0 | 0 | 0.5 | 1 |

*Table 10: Subtitling requirements fulfilled by SDL Trados Studio 2014*

Table 10 shows that the CAT tool SDL Trados Studio is, in terms of points, just as powerful as WinCAPS: both having achieved 4.5 points. The points, however, apply to different areas of usage; specifically, those that serve the tool’s initial purpose. While a TM, a term base, and linguistic quality checks are fully integrated in the CAT tool, a template, as one would need it for subtitling, is only conditionally possible; however, it can be expanded by using the tools that the software offers, such as characters-per-line settings. A thesaurus is available as a free add-on, while MT is only applicable after a paid subscription. ASR, auto-cueing, music identifier and OCR are obsolete due to the lack of a video player function. Without this context, no professional subtitle translation is possible, but merging the two tools (WinCAPS and SDL Trados) could result in a powerful subtitle translation tool.

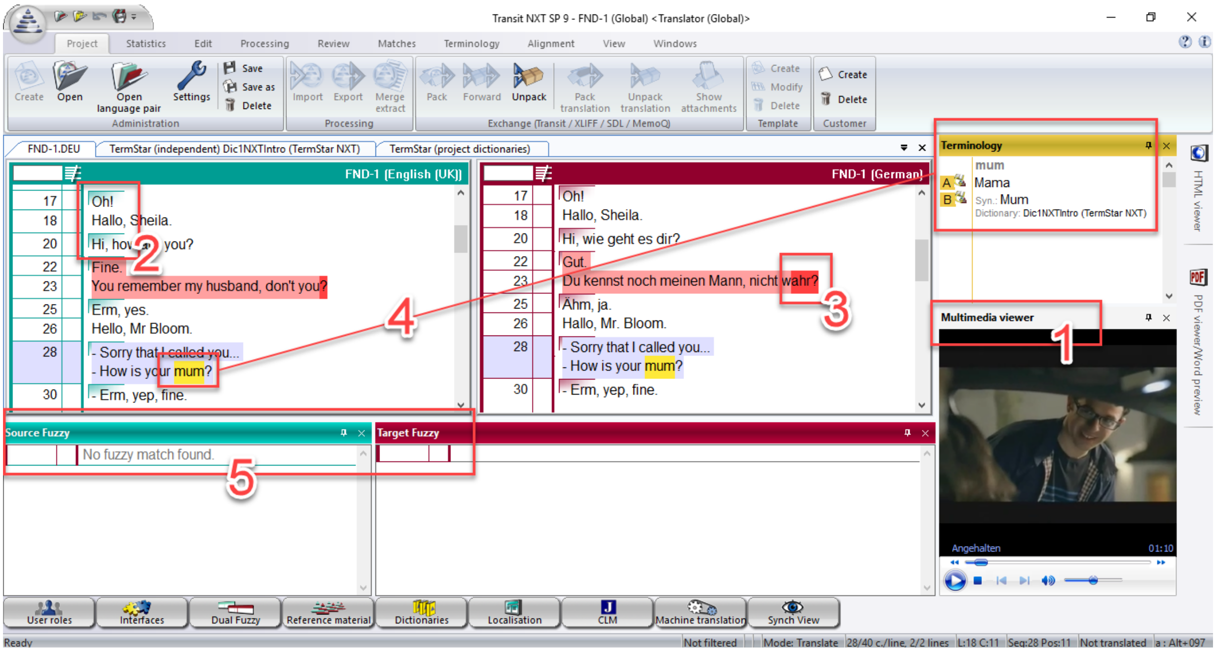
### 3.3.2. STAR Transit NXT

**What is it?** – STAR (Software Translation Artwork Recording) AG was founded as an LSP in 1984 in Switzerland and has been one of the forerunners in the development of CAT tools. Today, STAR Group runs its offices in 51 locations in more than 30 countries including UK, Finland, Thailand, China, Egypt, and Iran, just to name a few.[17] STAR Group offers translation and localization, terminology management, quality checking and proofreading, and MT solutions to individual translators and companies on a quarterly, semi-annual, or yearly paid licence basis. The translation tool is called Transit NXT with different packages available. For a freelancer, a 3-month licence costs €75.00 (as per 2017-07) and includes a TMS.

---

[17] For more see www.star-group.net/en/

**Subtitle file translation possible?** – Transit supports the file format SRT and adds the possibility to play the video simultaneously. The video has to be stored in the same folder and with the same name as the source file. Only the extension should differ; for example, video.mp4 and video.srt. When the files are loaded in the application, the video can be viewed in an extra Media Viewer window (as shown in #1 in Figure 11). Once the user clicks on a subtitle, the player plays the video from the subtitle's in time automatically. The user can also control the player manually for context. #2 in the figure below shows that the program carries out segmentation by subtitles, marked through an angle in the upper left corner, and also by sentences, which are denoted by line numbers. Any characters that exceed the SRT file's settings are marked in red as #3 shows. Identified terminology, which was previously saved to the respective dictionary, is marked in yellow (#4). TM matches are shown in two separate windows for source fuzzy and target fuzzy matches (#5).



FND-1 (English (UK))

17 Oh!
18 Hallo, Sheila.
20 Hi, how are you?
22 Fine.
23 You remember my husband, don't you?
25 Erm, yes.
26 Hello, Mr Bloom.
28 - Sorry that I called you...
- How is your mum?
30 - Erm, yep, fine.

FND-1 (German)

17 Oh!
18 Hallo, Sheila.
20 Hi, wie geht es dir?
22 Gut.
23 Du kennst noch meinen Mann, nicht wahr?
25 Ähm, ja.
26 Hallo, Mr. Bloom.
28 - Sorry that I called you...
- How is your mum?
30 - Erm, yep, fine.

Terminology: mum
A Mama
B Syn.: Mum
Dictionary: Dic1NXTIntro (TermStar NXT)

Source Fuzzy: No fuzzy match found.

*Figure 11: UI of STAR Transit NXT with an SRT for translation (SRT source: Channel 4's* Friday Night Dinner *S01E05, 2011–)*

**Quality levels** – Although I am familiar with using CAT tools during translation and therefore feel knowledgeable about them, the applications within Transit NXT were not straight forward. To import a TMX, I needed to search for instructions on the internet. It took some time to find out how to efficiently check for quality, and I failed to find any text-specific settings, such as character limitation per line. The settings were automatically taken by the program from the SL file. Furthermore, the folder structure for projects was not as clear as for SDL Trados: the files created by the software were not intuitively tagged and filed in the folder as there were several MTX, BAS and BAK files, the purpose of which I was unsure. The files with the extensions DEU and ENG were apparently the SL and TL files, but as opposed to SDL Trados, the user cannot simply open the bilingual file via the folder. The user consequently always needs to open the application first and only then select the respective project. Therefore, the translator must proceed through every step for each individual project rather than just open a file and work from there in the editing mode.

**Useful? –** For interlingual translation with a transcript in the SL, the tool possesses useful features in regards to subtitling insofar as it offers context thanks to a multimedia window, intelligent marking of segmentation, and built-in MT options. For intralingual translation, as always, a template is needed since there is no ASR integrated; additionally, adjusting time-coding or adding subtitles for sound description is not possible. The results are collated in Table 11:

| | **ASR for auto-transcription** | **Auto-cueing** | **Music/intro/ lyrics identifier** | **TM** | **Term base** | **Template** | **Thesaurus** | **Auto-colouring or tagging** | **OCR open subtitles** | **MT** | **QA** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Intra** | X | X | X | X | X | X | X | X | X | X | X |
| **Inter** | | X | | X | X | X | X | | X | X | X |
| **NXT** | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0.5 |

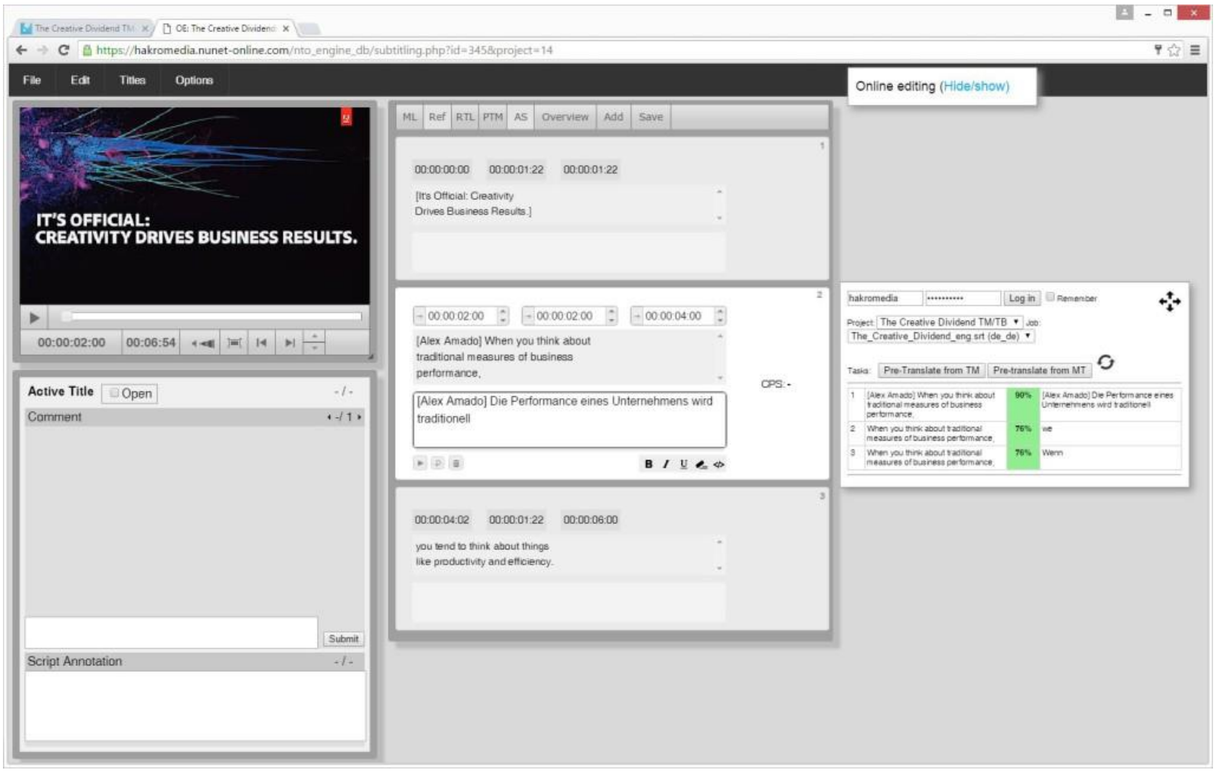
*Table 11: Subtitling requirements fulfilled by STAR Transit NXT*

Although the software does not offer auto-cueing, but instead the very useful synchronised video playing (which could be understood as cueing within the CAT tool) I will allocate the software 1 point since this feature deserves some acknowledgment. The TM, term base, and MT are all built-in, for which the tool receives 1 point each. The linguistic checks (QA) do not entail subtitling specific settings and score only 0.5 points. The resulting score is 4.5 points for STAR’s Transit NXT.

### 3.3.3. Hakromedia subCloud connected to Memsource

**What is it? –** Hakromedia GmbH is a Munich-based media localization provider for video, audio, and animation that has offered translation, subtitling, transcription, dubbing, and voice-over (among other services) since 2008. In 2016, Hakromedia connected their browser-based subtitle-editing workbench, subCloud, to the cloud-based CAT tool provider's translation workbench, Memsource, in a software as a service (SaaS) model. Memsource is a CAT tool, which is free for translators and offers web-based as well as offline workbenches. SubCloud offers a pay-by-use model and charges €4-6 per minute of the subtitled video, depending on the language.

**Subtitle file translation possible?** – The online editing module subCloud 'allows users to create and edit subtitles in the browser', while they 'are displayed in real-time against the streaming video' (Memsource 2016). The subtitler can create, translate, and edit subtitles online using a TM, term bases, and MT. An upload of existing transcripts is possible, and the service of segmentation and time-coding is included. QA is customisable, language-specific differences are taken into account, and all subtitle formats are supported. The tool serves, furthermore, as a project management tool with job chains, assignment, and workflow control.



https://hakromedia.nunet-online.com/nto_engine_db/subtitling.php?id=345&project=14

00:00:02:00 00:06:54

1
00:00:00:00 00:00:01:22 00:00:01:22
[It's Official: Creativity
Drives Business Results.]

2
00:00:02:00 00:00:02:00 00:00:04:00
[Alex Amado] When you think about
traditional measures of business
performance,
[Alex Amado] Die Performance eines Unternehmens wird traditionell
CPS: -

3
00:00:04:02 00:00:01:22 00:00:06:00
you tend to think about things
like productivity and efficiency.

hakromedia ••••••••••
Project: The Creative Dividend TM/TB Job: The_Creative_Dividend_eng.srt (de_de)
Tasks: Pre-Translate from TM Pre-translate from MT

| | | | |
|---|---|---|---|
| 1 | [Alex Amado] When you think about traditional measures of business performance, | 90% | [Alex Amado] Die Performance eines Unternehmens wird traditionell |
| 2 | When you think about traditional measures of business performance, | 76% | we |
| 3 | When you think about traditional measures of business performance, | 76% | Wenn |

*Figure 12: UI of subCloud (source: Memsource 2016)*

Since the usage of this tool needs to be authorised by companies, meaning that a subscription is needed, I was not able to test and evaluate it. Hence, no table of fulfilled requirements can be provided. The company's information revealed that the set-up of a new user to subCloud occasioned high administrative costs as this is a powerful tool and is subject to the number of projects, languages, and users that need implementation.

## 3.4. Summary: How will a CAT tool help in subtitling?

Taking all the arguments mentioned above into account, it becomes obvious that subtitling is an ever-growing market. Not only are the accessibility guidelines in many countries responsible for this, but ‘big data’ and the developments due to globalisation and internationalisation also play an important part in this growth. Today, we are producing data at an exponential rate. Thanks to freedom of movement and higher levels of education among westernised societies, people learn to speak foreign languages and move abroad. The film industry in general, and international film festivals in particular, strive to make their films accessible to their likewise international target audience and therefore need subtitles on their DVDs and cinema screens. According to the WHO, 285 million people[18] are estimated to be blind or have low vision and 360 million people have disabling hearing loss: 32 million of these are children.[19] The more data are produced on an international scale and the more people there are who make use of such data, the higher the demand is for software programs that facilitate work, accelerate productivity and, simultaneously, reduce costs.

There is the continuing argument that translation memories are not useful in an artistic discipline such as subtitling. Due to time and space constraints, utterances need to be shortened, re-phrased, tweaked in their meaning; thus, the translation or SDH might not fit into the context of a different show or episode. People working with this argument can be proven wrong with the simple counter-argument that professional translators would not be advised to accept a proposed fuzzy match without re-assessing it anyway. Furthermore, the industry and need for subtitling has outgrown the limitations of settings such as films and series. All sorts of content in all sorts of domains now demand subtitling.

When watching TV with subtitles, I have often noticed that there is a discrepancy between a transcript of an episode and its subtitles. This is due to time and space constraints that ensure that the reading speed does not exceed 140-150 words per minute in German. Even for movies shown in cinemas, the subtitles can differ from the subtitles for the same movie on DVD. The reason for this can be the aspect ratio of the image, which can differ between that for cinemas and that for traditional TV sets or DVDs. The aspect ratio of an image defines the proportion of its width in relation to its height; for example, 16:9, 4:3, 5:3, and so on. And, while in cinemas the aspect ratio is currently 2.35:1, the most common ratio for current HD and digital broadcast TV sets is 16:9 (resulting in a ratio of 1.78:1). This

---

[18] WHO 2014. http://www.who.int/mediacentre/factsheets/fs282/en/

[19] WHO 2017. http://www.who.int/mediacentre/factsheets/fs300/en/

means that there is more space for subtitles in cinemas than on DVDs, due to the wider screen. This simple example demonstrates that subtitles need to be translated or altered for different settings.

Clearly, subtitles differ not only between intralingual and interlingual forms, but also within one form itself. Let us take this example as a case in point: in the US comic series *The Simpsons,* there is a show called ‘Eye on Springfield’. This show appears in many different episodes and is always called ‘Eye on Springfield’ throughout the US version. In the German translated version, however, it has the following different titles (Table 12):

| Episode | German title | German title in subtitle |
|---|---|---|
| **S03E10** | Alle Augen auf Springfield | Springfield im Visier |
| **S04E06** | Augenzeuge in Springfield | Spot auf Springfield |
| **S04E20** | Brennpunkt Springfield | Brennpunkt Springfield |
| **S05E09** | -none- | Eye on Springfield |
| **S05E16** | Brennpunkt Springfield | Blickpunkt Springfield |
| **S08E17** | Ein Auge auf Springfield | Springfield im Rampenlicht |
| **S14E09** | Ein Auge auf Springfield | -none- |

*Table 12: Various translations of ‘Eye on Springfield’ within* The Simpsons

Thus, if you take these examples into consideration, even subtitles from the same movie or series can be different. But should a title in a movie always have the same translation? The rule of consistency is also valid in an artistic environment: even more so in an educational and commercial environment. A company’s slogan should not be altered, and neither should the wording of, for example, Newton’s laws always be the same and not vary in its translation.

# 4. Solution: Which CAT features should the AVTEnT have?

We analysed the CAT tool functionalities of current subtitling software in Section 3.2 and the subtitling features in current CAT tools in Section 3.3. Thanks to the analysis, it has been possible to identify several useful items that a merged subtitling software with CAT tool elements should be offering. The next step, thus, is to experiment with more tools that are available and blend their features into one powerful tool. The following section, therefore, contains field tests of the individual proposed functionalities where deemed necessary for the purpose of this thesis (thus focusing mainly on subtitling-related features) and a proposed design for a process of the same functionalities within the AVTEnT step-by-step during subtitle translation.

## 4.1. Optical character recognition and automatic speech recognition

The main issue of designing a CAT tool for subtitling environments seems to lie within the fact that the source text is a video with an integrated audio track, whereas the needed output of the target text must be in written form including time indicators that can be processed by subtitling software but should not cover existing text visible to the audience (such as open and integrated subtitles). The next sub-section, therefore, discusses the possible features of OCR (optical character recognition) and both speaker-dependent and -independent ASR (automatic speech recognition) for speech, music, and sound effects as well as auto-colouring or tagging of speakers.

### 4.1.1. OCR

In Chapter 3.1.1 (g) I identified the fact that in order to tag foreign language utterances without covering possible open subtitles, an OCR could be useful. As explained in *1.1 Computer-aided translation tool (CAT tool)*, OCR software[20] scans an image (i.e. the scanned version of a hard copy) for any text contained in it and identifies characters and words. The identified words are then compared to patterns stored in the software's database, so that matches are output and converted into editable text. The result is a text ready to be processed by a word processor such as Editor or Microsoft Word. This technology is very useful for translators, as they can quickly process a hard copy or PDF and feed it into their CAT tool, which saves time and lowers the risk of a strain injury as compared to having to type it in (see Bowker 2002). Some factors may affect the output quality, such as faded or blurred text, external disruptions to the text like a coffee stain, or the use of different fonts in one document (Bowker 2002: 26–27). Also, words in lines that are not straight, such as in a photograph of pages, can cause distortion. This can lead to letters being misinterpreted as numbers (S mistaken for 5) or confused with a different letter (the German ß for B). More sophisticated tools, however, for instance the pdf converter by Adobe, even recognise low quality documents' character patterns and set them in the context of a given language, so the results are based on contextual decisions rather than on isolated single characters. This kind of conversion technology means that there must be a way to recognise open subtitles so as to not cover them with closed subtitles.

---

[20] Read more on this in Bowker 2002: 22-30.

### *Proposed process for OCR*

When a subtitle file is newly created, with or without a transcript (speech-to-text alignment vs. auto-transcription) to hand, the video file will be scanned by the software to identify open subtitles. Once identified, the time codes are blocked for a preset period of time, usually 1–2 seconds, so that the user cannot enter a closed subtitle. The only problem to be tackled here is to make sure the OCR ignores text on screen in the background. Therefore, only intertitles, open subtitles, and integrated subtitles should be identified, and any other text must be ignored. To make sure that the user benefits from this functionality, an option should be built in to ignore or discard a blocked open subtitle.

### 4.1.2. ASR

The next issue identified is that the output required for the target text must be in written form, including time indicators. Hence, the subtitler needs to convert an audio file into written text and spot it. If it is possible with optical character recognition, the newest technology should make an aural recognition viable as well. This leads me to think of speech recognition software, as for instance Dragon® NaturallySpeaking.

Speech recognition is also known as ‘voice recognition’. Speech recognition software analyses the speaker’s acoustic input via microphone and breaks the sounds down into phonemes, which are then compared against stored word patterns in a database (Bowker 2002: 30) and written down automatically within a word-processing application. Two types of speech recognition programs are relevant for the purpose of this thesis: speaker-dependent and speaker-independent systems (Bowker 2002: 31–33).

For speaker-dependent systems, there is a vocabulary set already stored in the database that comes with the system, but the software allows users to add custom words or to respeak existing words in order to achieve a faster recognition. Speech recognition software works with strings of words and context for a more precise output of e.g. homophones. If the output does not match what was intended to be written, the words can be edited via keyboard, shortcuts and/or verbal command. To avoid editing, the software can be trained by the speaker. For that to happen, the speaker has to read a text provided by the software to generate a speaker-based vocal profile that can only be used by the selfsame person. In fact, some systems are so highly sensitive that the speaker must always use the same microphone, must not have a cold, since this changes their vocal patterns, and should not be surrounded by noise that could distort the sound of their voice, such as a telephone ringing in the background or car noises from outside.

In contrast to this, with speaker-independent systems, several different people's vocal patterns are collected and fed into in the database by the manufacturer to train the program and thereby raise the matching function independently of any given speaker. This is what Google offers with their huge database from Google Search and Google Translate, as do other applications that users can control by using their voice.

Speech recognition seems to be a useful tool, but the question poses itself: How reliable is its output? While speaker-dependent speech recognition systems need be trained by a single given speaker, Google has collected data from all over the world with various accents, linguistic variants and dialects. Newly conducted R&D reports of Screen Systems point to positive results with similar solutions to those of Google's, such as automatic transcription and time-coding of a script with Screen's 'WinCAPS Script Extractor', with the condition being that the user feeds an existing script into the tool. Unfortunately, this utility is not yet a commercially available product (confirmed by email as of 2017-06-23), but a flyer can already be accessed on their website.[21]

For the purposes of this paper, I investigated the following: Google's YouTube platform with its automated subtitling and spotting feature, comparing it with Screen's WinCAPS in combination with Dragon® NaturallySpeaking, and as a result, formulate a rule for ASR at the end of each field test.

### 4.1.3. Field test (1): YouTube's automatic captioning and auto-cueing

To carry out the field test on Google Translate's automatic subtitling skills, I uploaded four videos of one person reading from the book *Through the Looking Glass* by Lewis Carroll to YouTube. The source text for both languages is shown in Appendix 2, Table 17:

I recorded the videos on my phone with an MP4 output format:

**DE_clear**: one person reading the translated German version by Christian Enzensberger with good sound quality

**DE_poor**: one person reading the translated German version by Christian Enzensberger with poor sound quality

**EN_clear**: one person reading the English version in a clear manner

**EN_poor**: one person reading the English version with a thick foreign accent

---

[21] See flyer at http://subtitling.com/downloads/script-extractor-flyer/?wpdmdl=7468 (last accessed 2017-06-21)

### *Speaker-independent auto-transcription*

You can upload a video file of any format to YouTube. I clicked the *upload* button next to the profile picture in the upper right-hand corner and dropped the video files that I would like to have auto-captioned into the corresponding field on the YouTube screen. The average upload took 11′ 44″ for an average video length of 1′ 25″. Within *Advanced settings*, the *Video language* can be set to *German*, which results in the autogenerated subtitles being available only in German. It can take up to half-an-hour for the subtitles to become visible in the uploaded video if the following applies: poor sound quality, complex audio in the video, or lengthy video. Once the subtitles have been generated, the icon *CC* (see Figure 13) appears to be clickable within the editing view of my uploaded videos.

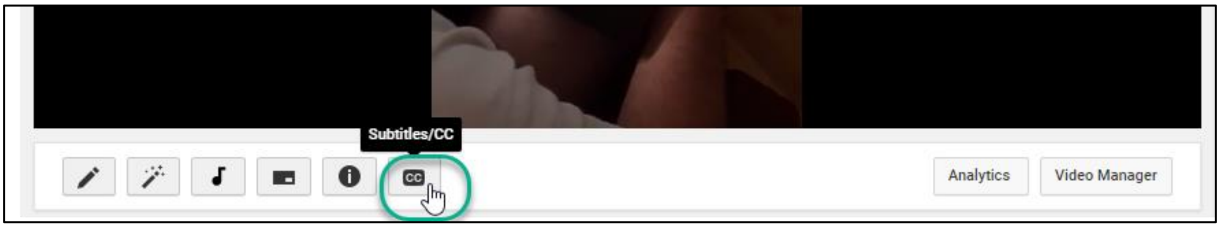


*Figure 13: Editing view of owned video on YouTube*

I clicked on it to get to the section *Manage subtitles and closed captions*, and clicked the button *German Automatic*, as shown in Figure 14:

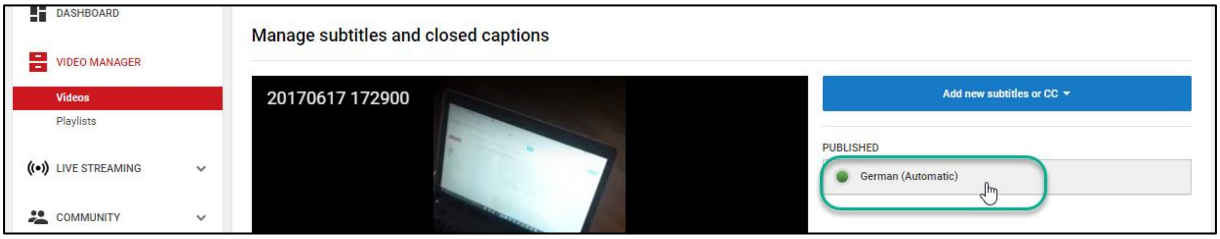
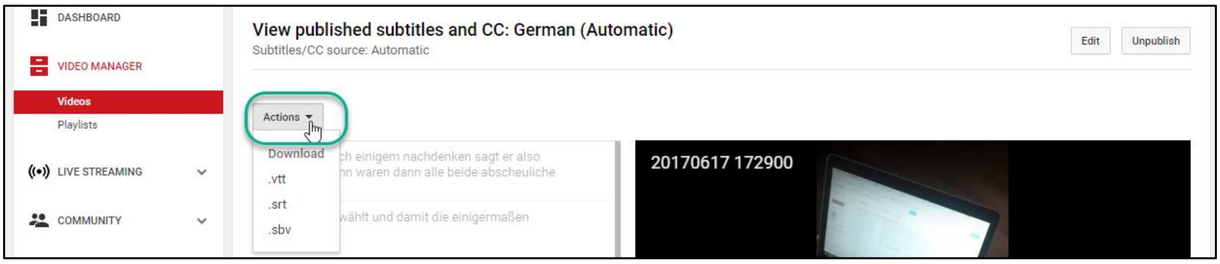


*Figure 14: Section* Manage subtitles and closed captions *to get to the list view of auto-generated subtitles*

I got to a page where I could download the subtitles file in the formats .srt, .vtt, or .sbv, or edit the subtitles directly within the browser (see Figure 15):



ch einigem nachdenken sagt er also
nn waren dann alle beide abscheuliche

wählt und damit die einigermaßen

*Figure 15: Download and editing mode for auto-generated subtitles including time codes*

Table 13 below shows the automated speech recognition outcome (hypothesis) of DE_poor as compared to the original text (reference). To be able to compare the effort required for each task, I opened the output in WinCAPS for post-editing (task duration 11′ 54″ for German, 06′ 54″ for English) and created the following subtitles, without complying with any guidelines or best subtitling practice (three lines, no semantic grouping, up to 313 wpm, no minimum gaps, overlaps ignored etc.) to be able to calculate the word error rate (WER). The WER calculation is one method used to evaluate the quality of speech recognition output. It is calculated by dividing the total number of instances of deletions (DEL) from the reference, insertions (INS) into the hypothesis, and substitutions (SBST) made in the hypothesis by the total number words in the reference (see second column). Thus, it shows what percentage of words were misheard.

| ORIGINAL (REFERENCE) | WORD COUNT | AUTO-TRANSLATION (HYPOTHESIS) | WORD COUNT | DEL | SBST | INS | ERROR TOTAL |
|---|---|---|---|---|---|---|---|
| Nach einigem Nachdenken sagte Alice: "Also, | 6 | nach einigem nachdenken sagt er also | 6 | | 2 | | 2 |
| dann waren sie eben alle beide abscheuliche | 7 | dann waren dann alle beide abscheuliche | 6 | 1 | 1 | | 2 |
| Gesellen", und damit hielt sie einigermaßen | 6 | gewählt und damit die einigermaßen | 5 | 1 | 2 | | 3 |
| bestürzt inne, denn sie vernahm ein | 6 | [Musik] | 1 | 6 | | | 6 |
| Geräusch, das ihr wie das Schnauben | 6 | | 0 | 6 | | | 6 |
| einer großen Lokomotive vorkam, ganz nah im Wald, | 8 | einer kurzen lokomotive vorkam ganz nahe | 6 | 2 | 1 | | 3 |
| und zugleich fürchtete sie, es könnte | 6 | zugleich führt er schmeißt anschauen | 5 | 1 | 3 | | 4 |
| vielleicht von einem wilden Tier kommen. | 6 | | 0 | 6 | | | 6 |
| "Gibt es in dieser Gegend viele Löwen und | 8 | | 0 | 8 | | | 8 |
| Tiger?", fragte sie zaghaft. | 4 | | 0 | 4 | | | 4 |
| "Das ist nur der Schwarze König", sagte | 7 | | 0 | 7 | | | 7 |
| Zwiddeldei. "Er schnarcht." | 3 | | 0 | 3 | | | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| "Komm und schau ihn dir an!", | 6 | | 0 | 6 | | | 6 |
| riefen sie zu zweit, fassten Alice beiderseits an den | 9 | klicken sie zwei fakten es beißt an | 7 | 1 | 5 | | 6 |
| Händen und führten sie zu dem schlafenden König. | 8 | händen und füßen den klassen könig | 6 | 2 | 3 | | 5 |
| "Sieht er nicht wunderhübsch aus?", fragte Zwiddeldum. Da hätte Alice freilich lügen müssen. | 13 | 150 ausrichten gesungen da hätte er | 6 | 11 | | 2 | 13 |
| Er hatte eine hohe schwarze Schlafmütze | 6 | einen hohen eine hohe straße schlafen musste | 7 | 2 | 2 | 2 | 6 |
| mit einer Quaste auf, lag zu einem | 7 | mit einer trage auf lag zu einem | 7 | | 1 | | 1 |
| unordentlichen Häuflein | 2 | unordentlichen häufleins | 2 | | | 1 | 1 |
| zusammengerollt da und schnarchte laut. | 5 | zusammengerollte am schlechte laut | 4 | 2 | 1 | 1 | 4 |
| | 129 | | 68 | | | | 96 |
| | | | | | | **WER** | **74%** |

*Table 13: German poor quality video - YouTube output compared to post-editing output in WinCAPS*

As the comparison shows, the WER for the auto-transcribed poor-quality file is 74%, meaning that only 26% of the content was recognised and transcribed correctly, which translates into a quarter of the video being correct. In comparison to this, the clearly delivered video DE_clear had a WER of only 31%, meaning, therefore, that two thirds can be considered correct. What do these two results mean for deaf and hard-of-hearing audiences, though? They mean that the quality produces results in which only one in four to two in three words are correct. Now, as a thought experiment, and to experience what that means, try to comprehend this paper by only reading every fourth word.

Nevertheless, the bottom line is that the technology behind this process is amazingly advanced and on its way to improving its output quality levels for each language individually. Table 14 shows the WER results for all four videos:

| Video DE | WER | Video EN | WER |
|---|---|---|---|
| DE_clear | 31% | EN_clear | 22% |
| DE_poor | 74% | EN_poor | 30% |

*Table 14: WER results of YouTube's ASR for all four videos: German & English clearly and poorly delivered*

The conclusion is that for the non-representative, once-only experiment, Google's ASR for English has a higher quality output than that for German.

### ***Auto-cueing***

Apart from calculating the WER, the purpose of this task was to time-code the text with the corresponding utterance and upload a post-edited subtitle file to YouTube to see how it handles the newly proposed subtitles for replacement. To do so, I clicked on *Add new subtitles or CC* (blue button in Figure 14*)* and clicked *Upload a File*. There are two options for files: *Subtitles file*, and *Transcript*. For the *Subtitles file* option, the following file formats are compatible: SRT, VTT, or SBV/SUB, LRC, CAP (for Japanese captions), RT, TTML and a few more (YouTube Help n.d.).

To upload a *Transcript*, the content should be saved in plain text format. In order to get this file format within WinCAPS, I clicked File/Export/Text/Plain text and uploaded the plain text file to YouTube. I selected *Set timings* (loading time for German 04′ 20″, loading time for English 01′ 11″). Once the timings were set, a download of the newly time-coded subtitle file with the extension *.sbv became available for download. I clicked *Save Changes* to get back to the section *Manage subtitles and closed captions* as shown in Figure 14. I tried to open the SBV file in WinCAPS, but it turned out that this file format is not supported, so I opened it with Editor and saved it as an SRT file by using *"[name].srt"*, but this file could not be processed either because it contained SBV-specific time code formatting which differs from that of a true SRT file, as explained further above in Figure 9. So, I used the online SBV-to-SRT converter from Gideon Goldberg[22] to be able to see the outcome of YouTube's auto-synchronising of the transcript with the audio compared to the simple post-editing task. Just the post-editing and time code adjustment task took 10′ 33″. The upload of the transcript and having time codes automatically set by YouTube took 01′ 27″.

Will the content be readable if I measured quality by the Ofcom's recommended maximum reading speed of 160-180 words per minute for English? Table 18: Evaluation of

---

[22] See https://gidsgoldberg.com/sbv_docs_converter.html

Auto-synchronization using wpm as a factor (see Appendix 3) shows a comparison of automatically synchronised subtitles by YouTube compared to manually time-coded subtitles in WinCAPS and their respective wpm rates. Both outputs had only one subtitle with a high reading speed, but the auto-synced output by YouTube was still within limits and lower (171 wpm) than the quick manual cueing (201 wpm). The low WER for the English files and the good quality German files and the low reading speeds for automatically cued English files are evidence of the excellence of YouTube's function as compared to human typing and spotting. In addition to the faster and more accurate output as regards for reading speed, the online application includes automatic sound descriptions for e.g. applause, music, and laughter (Wang 2017: online). Consequently, I will consider this functionality very useful for my AVTEnT, especially for intralingual subtitling.

#### ***Proposed process for ASR***

So, what should the ASR be capable of doing in a subtitling environment? When a subtitle file is newly created, with or without a transcript (speech-to-text alignment vs. auto-transcription) to hand, the video file will be scanned by the software to identify speech sequences, shot changes and sound effects, to then transcribe the input, while at the same time fixing in and out times according to the preset values. The output of the audio/video input will be an auto-transcript including sound descriptions with automatically set time codes in a post-editing module. During auto-transcription and auto-cueing it, the software should observe such settings as minimum gaps, wpm rates, shot changes, lines per subtitle and characters per line, which should be manually adjustable. If this is not possible due to excessive text, the software should automatically highlight subtitles that require post-editing. The post-edited segments will be stored in the database in alignment with the audio track in order to train the ASR.

### 4.1.4. Field test (2): WinCAPS and Dragon® NaturallySpeaking

To test ASR in a subtitling environment, I tried WinCAPS in combination with the speaker-dependent and licensed speech recognition and dictation software *Dragon® NaturallySpeaking 13* for German-language dictation, which Screen Systems Ltd offers in a package for the paid-extra option Q4 Pro. Screen states on its website that

> "Good speech recognition software for live captioning is not just about reasonably accurate dictation […] but also requires real-time, low-latency processing and essential tools such as vocabulary management and speaker training to improve accuracy. It's worth considering that an

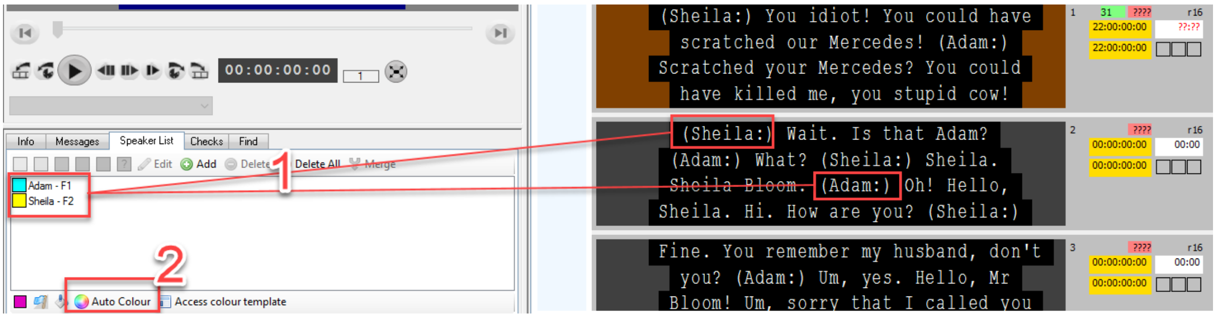
> accuracy of 90% may sound great, but in practical terms that is an average of about one word out of place in every sentence!"
>
> (Screen Subtitling Systems n.d.: online)

Good speech recognition *for auto-transcription* is, consequently, even less realistic for the time being. This is because speech in scenes that for example contain arguing, overlaps, and background noises of cars, planes, trains, restaurant chatter, etc. can distort speech. Then again, for company promotion videos and scientific or e-learning videos, speech is usually very clear, so that a speaker-independent ASR application should be able to catch all words and phrases.

### *Speaker identification (colour or tag)*

Speaker identification by colouring or tagging speakers is particularly used in intralingual subtitling, but can also be required for interlingual subtitles, such as for voices in the off or for clearer speaker identification in fast-interchanging dialogue. Whilst WinCAPS has not so far incorporated auto-transcription into its products, it does include an auto-colouring function. In the Q4 Pro version, automatic speech-to-text alignment is also on offer. Both functions, however, only work via an import of a script. For auto-colouring, the script needs to contain speaker information which the user, after importing the script, needs to match with the respective speaker list in the toolbox (#5 in Figure 7). By clicking on the *Auto Colour* button, the subtitles get automatically coloured according to speaker.

The field test, unfortunately, did not produce the expected results. WinCAPS's Help menu does not point to any sort of formatting required for speaker identification. As a result, my imported script contained speaker denotation in brackets. Although I created a speaker list (as seen in #1 in the figure below), the *Auto Colour* button (#2) does not colour the speakers accordingly.



(Sheila:) You idiot! You could have
scratched our Mercedes! (Adam:)
Scratched your Mercedes? You could
have killed me, you stupid cow!

(Sheila:) Wait. Is that Adam?
(Adam:) What? (Sheila:) Sheila.
Sheila Bloom. (Adam:) Oh! Hello,
Sheila. Hi. How are you? (Sheila:)

Fine. You remember my husband, don't
you? (Adam:) Um, yes. Hello, Mr
Bloom! Um, sorry that I called you

*Figure 16: Unsuccessful auto-colouring test in WinCAPS (standard license) (text source: Friday Night Dinner S01E05, Channel 4, 2011–)*

The Help menu should state which formatting is needed for the feature to function properly. Adding brackets to the names in the speaker list as provided in the script also fails to produce results.

### *Speaker-dependent transcription and music identifier*

To test the usefulness of speaker-dependent speech recognition software within subtitling, i.e. for transcription, I used the dictation software Dragon® NaturallySpeaking for German to dictate the recap scenes of three consecutive German episodes of the American TV series *The O.C.* I chose German for two reasons: a) I would like to focus on intralingual subtitling of SDH as this is the rather challenging part for a CAT tool in subtitling environments, and b) Dragon® NaturallySpeaking is a licensed and language-specific dictation software, and I have a licence for the German language version.

For the experiment, I decided to use three different procedures: a) I entered subtitles from scratch via keyboard and spotted the file as I went along; b) I dictated a transcript into a word-processing application and fed it into WinCAPS to post-edit it there and set time codes manually; and c) I used the dictation software directly in the WinCAPS environment and set the time codes manually during the process. To compare the effort required for each process, I measured the time needed for each task and evaluated the outcome. For music identification, if needed, I additionally used Shazam and SoundDog, the music identification apps for Android smartphones and iPhones.

The test results are as follows:

| TASK | The OC S02E05 (28,76″) | The OC S02E06 (29,04″) | The OC S02E07 (28,20″) |
|---|---|---|---|
| **TSCR via keyboard** | 04:19:94 | | |
| **TSCR via Dragon, Word, Wincaps** | | 09:12:23<br>(of which 05:22:57 for transcription) | |
| **TSCR via Dragon within Wincaps** | | | 8:50:34 |
| **Editing down observing rules** | 10:41:51 | 09:52:73 | 07:20:87 |
| **Total task duration** | **15:01:45** | **19:04:96** | **16:11:21** |

*Table 15: Test results for editing tasks within WinCAPS Qu4ntum*

For procedure c, i.e. transcription via Dragon® within WinCAPS, an extra window for dictation opened every time I started speaking, for which I had to voice-command the software to click on the *Transfer (Übertragung)* button to enter the dictated text into the subtitle. Only then was I again able to press *Play* for the video. For spotting, I used the keyboard.

For procedure b, i.e. transcription using Dragon, a word processor, and WinCAPS, I encountered two problems: due to switching between two applications – the editor and the subtitling software – I wasted a lot of time, and, secondly, since I was transcribing the scenes in an external window, I had no control over the restrictions applied by the subtitling software. So, when I finished transcription in the editor, and copied the content into WinCAPS, the software automatically segmented it applying preset rules, such as characters per line (37) and number of lines per subtitle (two). An example is:

| **Rules: maximum 2 lines, 36 characters per line.** | | |
|---|---|---|
| **Source** | Ich wollte dir nur sagen, dass ich nicht mit dir reden will, okay? Ich komme damit klar. | |
| **Will be segmented according to rules to a** | | |
| **Target** | Ich wollte dir nur sagen, dass ich | (34 characters) |
| | nicht mit dir reden will, okay? Ich | (35 characters) |
| | komme damit klar. | (17 characters) |

*Figure 17: Example of automatic subtitle segmentation applying preset rules*

As shown in Table 15, the fastest way to transcribe the recap scenes of around 29 seconds was using the keyboard, even though editing down to the language- and customer- specific time and space constraints, took the longest time. Transcription of the scenes using Dragon® NaturallySpeaking in the WinCAPS environment required the shortest amount of time in regard to editing down, and was faster than using the dictation software within a different application and then copying the contents into the subtitling environment.

For SDH processes, I use Shazam and SoundHound to identify songs and to find their lyrics. To put the required information into the subtitle, I needed to follow these steps:

1) stop the video and play back to the beginning of the song,
2) take out my smartphone,
3) open the Shazam or SoundHound app,

4) press *play* in the subtitling software and at the same time activate Shazam,
5) write down Shazam's instant result in Google to get the lyrics,
6) press *pause* in the subtitling software when the song finishes to see how much of the lyrics has been sung, and
7) fill in singer and song title, copy lyrics into the subtitling software, and colour as well as spot all the information as per style guide.

These cumbersome seven steps could be avoided if the software did all of this on its own when scanning the file for the proposed ASR transcription. The ASR should also be able to identify theme songs and time-code them automatically in a consistent way throughout all episodes of a series/show/course.

### ***Proposed process***

To resolve the problems described above, I suggest the following method. When a subtitle file is newly created, with or without a transcript (speech-to-text alignment vs. auto-transcription) to hand, the video file will be scanned by the software to identify speakers, music, and sound effects to then colour and/or tag them automatically, if the user or a customer template tells the software to do so. Working with a transcript, the software should be able to identify and auto-colour speakers easily with the option to correct possible identification failures during the process of subtitling.

For SDH, the ASR should store samples of each actor's voice and, consequently, identify speakers accordingly. This means that in compliance with a possible customer template or the user's commands, the software should auto-tag or auto-colour identified speakers on screen. If the user works on an SDH file without a transcript, the auto-recognised elements should be stored in a TM in the following two categories: the auto-transcribed and post-edited version in the SL segment and the edited-down subtitle that fits the required wpm, number of lines, and characters per line settings in the TL segment. This is important, as the ASR needs to know the exact sound of the utterance and the TM needs to know how this utterance was translated so as to suggest the segment as a fuzzy match in the future.

With regard to music recognition, the software should look up the song and its interpreter plus the lyrics on the web and write into the segment that a song match was found. When the user clicks on the segment, a comment box should then open with information on interpreter, song title, and lyrics. That way, the user can decide whether to write down singer and song title and whether to subtitle the lyrics. If lyrics are required, then the user can select, using the mouse, how much of them should be copied in. Since at this point the video has already been scanned for transcription, the software then immediately observes wpm rates, minimum gaps, and shot changes and cues the song information and lyrics accordingly. If the cued and transcribed lyrics are too long for the preset rules, the software should highlight instances, so that the user can check up on them and alter them manually. In cases where the song could not be identified, e.g. when the song was only created for the context of the video and is sung by one character (for instance in the series *New Girl*), there should be an option to machine-translate the input. This would be useful in such series as *The Simpsons* or series for children with translated theme songs, but also in e.g. science-related YouTube videos for children that involve singing. Rules on how to tag music should be stored in a template.

### 4.1.5. Summary: OCR and ASR in AVTEnT

Looking at the rules identified below in this summary, it is clear that ASR in AVTEnT is a useful and needed tool, yet improvements need to be made to achieve a lower WER rate and a broader area of usage, such as for unclearly delivered speech. YouTube's speaker-independent auto-transcription and auto-cueing tool is powerful and should be implemented in a future AVTEnT. WinCAPS is already a sophisticated tool for subtitling, but could use some more features that not only function properly, but are also meticulously documented in the Help menu:

An integrated **OCR** feature should be able to identify intertitles, open subtitles, and integrated subtitles. Any other text must be ignored. This is to help the translator or subtitler to not cover them by closed subtitles, as best practice recommends.

The **ASR** should have a WER rate of less than 25 per cent (one word in four is misunderstood) and perform auto-transcription including automated cueing in compliance with minimum gaps, line length, wpm and character per line rates. Furthermore, it should observe shot changes and automatically fix out-times before a shot change. If compliance is not possible, the software should display the subtitles that exceed any preset limits. The use of a speaker-dependent dictation software should be supported by using e.g. an integrated relation to the software licence.

For SDH, the user should be able to train the ASR tool by feeding it with a few episodes to either a) recognise the speakers and colour them automatically or b) tag the speaker if no colouring is used. As for music identification, a tool such as Shazam, which identifies music chords, should be included. It should look up the song and find the lyrics. For the process to be clearer, the user ought to be able to decide how to handle songs and their lyrics: whether or not to write down the interpreter plus song title and whether to write the lyrics in full.

All the above leads to the conclusion that the user should be able to store information in a customer or TV show template containing the following rules: whether to auto-colour or auto-tag identified speakers, how to handle music and lyrics, and how to handle speech that crosses shot changes.

Post-editing thanks to a lower WER rate should be viable in a shorter period of time and both text and time codes should be adjustable. Since Dragon® Naturally Speaking states that their recognition rate is 99–100 per cent, more research should also be done in the field of speaker-*in*dependent ASR.

## 4.2. Machine translation (MT)

MT still needs to be developed further to reduce the cognitive effort of human intervention in post-editing steps. Output quality can be measured by *word error rate* (WER), *bilingual evaluation understudy* (BLEU), or *translation edit rate* (TER), but also by review feedback from translators (Declercq 2015: 485). Just as in the SUMAT project, where the online dictionary Linguee.com and Google Translate are both corpus-based, a high volume of bi-directional translated reference subtitle files will be needed in order to have MT in subtitles. The SUMAT project resulted in a cloud-based service that offered users the ability to upload their subtitle files and get a machine-translated file for download in seven languages. The service has been taken down, so the website is not available anymore (as of 2017-06-18). The identified subtitle translation platform FilmTit works with the same principle. My idea of MT in subtitling software is that once the audio has been automatically recognised and output as written text, this text can then be automatically translated by machine translation.

Wilss claimed in 1994 that MT would not provide useful assistance in translation since ‘die Programmierung des Rechners für das Übersetzen […] und die Postredaktion maschinell erzeugter Übersetzungen soviel Zeit in Anspruch nimmt, daß, genau besehen, zumindest derzeit der Übersetzer letztlich besser abschneidet als die Maschine. Dies gilt vermutlich auch für die Zukunft, gleichgültig, was clevere Programmierer noch alles an Programmiertricks erfinden werden’ (Wilss 1994: 169). He was convinced that post-editing would take too long for MT ever to replace translators, no matter what engineers might have up their sleeves. His prediction has proved true to date. But he also wrote that neither will it be possible in the future, no matter what clever engineers were to invent. In my view, this is not entirely true, since MT is about to get better thanks to post-editing, and since Google has collected a huge volume of data, I am confident that there is enough data to fill a corpus for MT of subtitles as well. Since ‘studies have shown that post-editing high-quality MT can, indeed, increase the productivity of professional translators compared to manual translation “from scratch”’ (Koponen 2016a: 132), I had to try it personally to find out if it is also suitable for the purpose of subtitling and also to time the overall effort. SDL Trados offers AdaptiveMT, and STAR’s Transit NXT has a built-in connection to iTranslate4.eu, Google Translate, MyMemory, and Microsoft Translator, all of which require an authorisation API key.

To refute Wilss’ assumption about the future, I looked into the newest freely available MT technology: YouTube’s auto-translation feature for auto-generated subtitles. Google’s YouTube works with the engine of Google Translate. Like any other corpus-based engine, Google Translate works like this: If you enter one word and add another, the whole search result changes due to there being a higher number of matches in context. For instance, *see you* will be translated into German as *wir sehen uns*. Add only one word, such as *see you* ***later***, the result will be changed to *bis später*. The same goes for individual letters: *arme Ritter* will be translated as *poor knights*, but with only one added letter *arme****r*** *Ritter* becomes *French toast*.

Google’s YouTube platform offers automatically generated subtitles in 10 languages: Dutch, English, French, German, Italian, Japanese, Korean, Portuguese, Russian, and Spanish. Not only do they offer ASR for automated transcription, but also automatic cueing (*auto-sync*), which stays on screen for long enough to read it (see results below in Table 18). YouTube, furthermore, offers automated speech alignment if the user provides a transcript, and auto-translation by MT into 51 languages in 2009 (Harrenstien 2009: online), which has been expanded to 100 languages (as of 2017-07-01). In addition, YouTube offers script extraction, the ability to jump to the point in the video that corresponds to a given subtitle, and the ability to manually import edited subtitles. One thing the automated transcription does not include is punctuation. There is no voice command function on YouTube for punctuation, which in general, for the time being, can only be added via speaker-dependent ASR software.

### 4.2.1. Field test (3): MT of YouTube

Since the official text of the translation by Enzensberger (1974) mentioned earlier is available online, I am curious to find out whether Google’s MT engine started to search through its huge database for corpus-based results. As mentioned in the previous section, YouTube offers not only auto-captioning, but also auto-translation of both auto-generated and manually prepared subtitles. The content of both versions, the original by Lewis Carroll *Through the Looking Glass* and the German translation by Christian Enzensberger *Alice hinter den Spiegeln*, can be found online.

Did Google identify the source to match the two in order to auto-translate from German into English? Surprisingly enough, the German content cannot be matched correctly with the English original, and therefore the auto-translation output is gibberish. Even more surprisingly, instead, German-Russian auto-translation is output in comprehensible Russian. Table 16 shows just one short extract for reference.

| German official translation | Russian official translation | Russian auto-translation |
|---|---|---|
| „Gibt es in dieser Gegend viele Löwen und Tiger?“, fragte sie zaghaft. | - А в вашем лесу много тигров и львов? – робко спросила она. | «Есть ли в этой области много львов и тигров?» предвратительно спросила она. |
| "Das ist nur der Schwarze König", sagte Zwiddeldei. | - Это всего-навсего Черный Король, - сказал Траляля. - | «Это просто черный король,» сказал Труляля. |

*Table 16:* Alice hinter den Spiegeln *vs. Russian translation by Demurova vs. YouTube's auto-translation into Russian*

The words that do not match can be translated in many ways, hence the differences. One fault is the incorrect translation of Twiddledee who should actually be called Tralyalya rather than the suggested auto-translation Trulyalya. Other than that, the syntax and punctuation are correct, even if it looks as if they are not. It is a relatively good output, but not quite what I would have expected from a machine translation tool as full of data as Google Translate is.

An extra feature, however, is the post-editing possibility for auto-translated subtitles. The process is as described in *4.1.3 Field test (1): YouTube's automatic captioning and auto-cueing* with a post-editing module online and the functionality to download the draft and post-edit it in a subtitling software or a TXT file.

### 4.2.2. Conclusion: MT in AVTEnT

As identified in Sub-section 3.1.2 (c), the use of MT will help to create fuzzy matches for subtitles that the TM may not contain. MT is gaining importance: this shows in academia through a rising number of studies on post-editing effort and MT outputs, but also in the translation industry with SDL Trados, STAR, YouTube, and many more providers offering MT in their solutions. After testing the tool in the field, I am more than convinced that this kind of feature will prove extremely helpful for subtitlers.

As mentioned before, MT focuses on the productivity side of the equation, i.e. cost reduction and profit increase for customers. My intention, in contrast to this, is to focus on how to help subtitlers and translators in their daily professional lives and reduce costs while increasing profit for the subtitle software users who also pay for the license. Therefore, I would like to propose a solution to raise quality and consistency ease for the subtitler and/or translator. As Guerberof suggests, a combination of a TM with MT will result in higher productivity than that of human translation from scratch, since an MT suggestions can be as helpful as an 85–94% fuzzy match from a TM (2014: 165). Following this suggestion, I will move on to testing a TM with all its beneficial features, such as alignment, concordance search, and QA.

## 4.3. Translation memories (TMs), alignment, and quality assurance (QA)

I would like to briefly refer to my findings as explained in detail in Section 2.2: Declercq's statement about previous studies coming to the result that 'using the TM even slowed down productivity by 2.5 per cent, whereas MT increased this by 24.5 per cent, a combined difference of 27 per cent or nearly a third' (2015: 486). Guerberof (2008) came to the conclusion that MT brings about a productivity gain of up to 25%, as compared to human translation from zero, while a TM propagates errors and therefore even decreases productivity by up to 8%. Bowker, in her quality study, concludes that 'translators using TMs may not be critical enough of the proposals offered by the system' (Bowker 2005: 19). In her 2014 article, Guerberof points out that 'translators […] using machine-translated output […] productivity and quality is not significantly different from the values obtained when processing fuzzy matches from translation memories in the range 85-94 %' (Guerberof Arenas 2014: 165) and that when a translator was using MT or a TM to translate sentences, these sentences had fewer errors.

In the same year, Teixeira concluded that 'the use of TM and MT are similar in the sense that both TM and MT provide suggestions that can help the translator or post-editor to generate viable solutions' (Texeira as cited in Koponen 2016b: 17). The developers of SDL Trados have reacted to this and made 'self-learning machine translation (MT) engines' (SDL Trados Studio 2017 Help documentation, online) available in the latest Studio version 2017, which delivers improved outputs of the MT by learning from post-editing by the translator. All of this confirms my findings in Sub-section 3.1.2 (b). To see if this is true for subtitles as well, the statement led me to experiment with the heart of any CAT tool: the translation memory.

Some companies like Wados in South-Korea, but also the worldwide operating Netflix, make use of TMs that are cloud-based, such as Sfera Studios used by Netflix, and subCloud used by Hakromedia. So, clearly there must be an advantage to doing so, but the question is:

What exactly is the advantage? And what is the situation for intralingual subtitling; how useful is a TM there? I consider a TM useful for many different scenarios:

**Series** – For consistency and time-saving within series that have repeatedly appearing punchlines, slogans or in-series shows, but also for previews, teasers for new episodes, and most importantly, series openings with recaps of what happened so far.

**Video game marketing videos** – with an ongoing story, such as *The Legend of Zelda*, *Final Fantasy*, but also *Professor Layton*, which is for children and adults alike. The translator saves time instead of looking through previous translations of reappearing sequences and flashbacks.

**Film adaptations –** Since the book upon which a film is based has usually been translated before being adapted for the screen, the book includes useful references for translation of the film. This does not only apply to single novels. A TM is possibly even more useful for film adaptions of sequels, such as for the names of magic spells in *Harry Potter*, or places and names in *Lord of the Rings*, or prequels that usually show what had happened in the past of an older film, just like *Star Wars Episode I – The Phantom Menace* from 1999 was the prequel of *Episode IV – A New Hope* from 1977. The TM here covers the need for two modes: an already existing translation of a book, and a translation of a series.

**E-learning** – For consistency and customer- or domain-specific terminology it is not only a term base that is useful:  a TM can also help. Slogans or repeated parts within sections of the e-learning course can be stored and reused.

**TV shows –** containing repeating mottos or recurring flashbacks can also benefit from the use of a TM.

Apart from their obvious usefulness in these scenarios, stored translation units within a TM have the advantage that the wpm rate and line length are already stored with the *language*-specific restrictions. However, reading speed, line length, and characters per line limitations differ from customer to customer, but also from show to show, depending on the speaker’s pace. Therefore, information on too long lines should be pointed out to the translator within the fuzzy match display: what is the wpm, the CpL and the context? For a better exchange of translation units among translators who work in a team, the subtitles should be storable in a TMX format for import and export into projects.

Here, my experiment begins: I wanted to test the usefulness of a TM in the setting of a recap of a series. Since I had already created some recap files during the testing of WinCAPS combined with speech recognition software, I reused them as a sample. To focus on SDH, I decided to take three consecutive German-language episodes of *The O.C.* In order for a TM to be applicable in an SDH setting, it was necessary to assume that ASR works with a maximum word error rate of 25 per cent so as to reduce my post-editing effort to one word in four and produce an acceptable transcript word-by-word. This word-by-word transcript serves as the SL in the TM.

To test of the TM, I used the output of my speaker-dependent transcription for the SL segments and the edited down subtitles as the target text and stored the segments in a bilingual TMX file, which I could later import into the CAT tool Transit NXT. Why did I choose Transit NXT? Because it is the only application that has a built-in video playing function, which provides me with context. This test was aimed to show me whether, for recap scenes, the software would suggest large numbers of relevant fuzzy matches. But first I had to align several subtitles in order to use them as a reliable source.

### 4.3.1. Field Test (4): Across's Alignment wizard

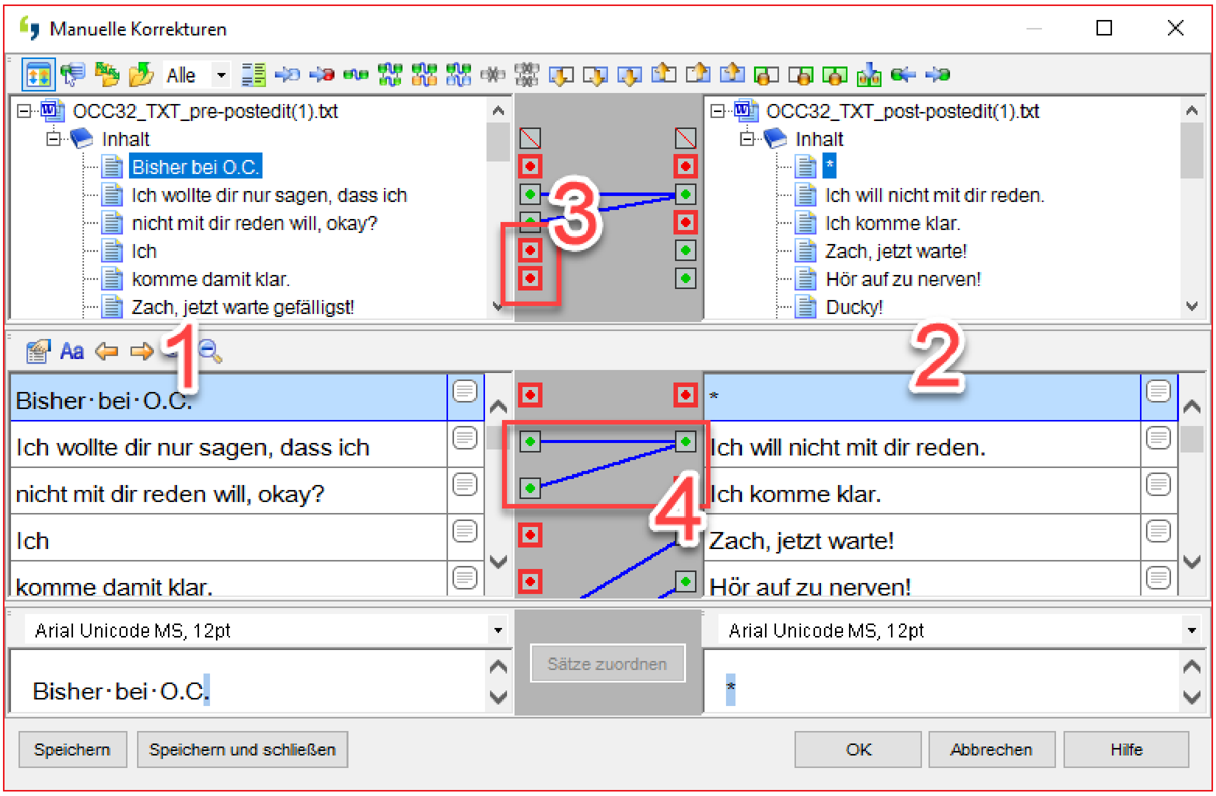
To create and expand a translation memory, the translator has more options than merely translating files and storing the respective translation units. Translators can also build up TMs by aligning translation segments and adding them to an existing master TM. When I need to align documents, I prefer to use the free version of Across[23] as it relates segments to one another automatically, but also allows for a manual cross-check by the translator. Other such tools are available within such licensed programs as SDL Trados Studio and Déjà Vu. Abbyy Aligner Online used to offer online alignment services, but has been taken down, so only a paid-for version can be purchased on their website.

Within Across, I simply clicked on *File/Alignment* to start the alignment wizard. I selected the word-by-word transcribed plain text file as source text with German-DE (German spoken in Germany) being the SL and then, in accordance with time and space constraints, post-edited plain text file as target text with German-AT (German spoken in Austria) being the TL. The output is shown in Figure 18. When the Across alignment wizard finished analysing the SL and TL files, I could select either to import or to manually correct entries. To make sure that the alignment had worked properly, I selected the manual correction option. This way, I was able to see the SL on the left-hand side (#1 in Figure 18 below) and

---

[23] The Across Systems GmbH was founded in 2004 to focus on the development of its software platform for translation management. For more information see http://www.across.net/en/company/about-us/

the TL segments in the right-hand column (#2):



OCC32_TXT_pre-postedit(1).txt
Inhalt
Bisher bei O.C.
Ich wollte dir nur sagen, dass ich
nicht mit dir reden will, okay?
Ich
komme damit klar.
Zach, jetzt warte gefälligst!

OCC32_TXT_post-postedit(1).txt
Inhalt
*
Ich will nicht mit dir reden.
Ich komme klar.
Zach, jetzt warte!
Hör auf zu nerven!
Ducky!

| | |
|---|---|
| Bisher·bei·O.C. | * |
| Ich wollte dir nur sagen, dass ich | Ich will nicht mit dir reden. |
| nicht mit dir reden will, okay? | Ich komme klar. |
| Ich | Zach, jetzt warte! |
| komme damit klar. | Hör auf zu nerven! |

Bisher·bei·O.C.

*

*Figure 18: UI of Across Alignment wizard (text source: The O.C. S02E05, 2003-2007, German: OC California)*

Any segments that have no match are marked with a red square containing a red dot (#3). Any segments that have been matched automatically are marked with a blue line that connects different segments which have green squares containing a dot (#4). Once I click on a red dot in the SL, a blue line appears which I can pull to a TL segment and drop it there. Now, the matched segments appear as green dots interconnected by a blue line. I can match SL with TL segments as one to many or many to one. This functionality is particularly important for subtitling as during subtitling some subtitles can be merged into one to accommodate wpm rates and line limitations; for example, if one out of two subtitles is far too long and impossible to shorten, it can be pulled together with the previous or next shorter subtitle, as long as the result does not exceed the maximum number of lines. To serve better in translation situations, it is necessary that all lines and segments that belong together are represented in one segment later on.

This means that you relate one SL sentence that was split up into two TL sentences. This way, the TM can be maintained while at the same time quality can be assured, which is one aim of this thesis: to help subtitlers raise their output quality.

When I finished correcting the automatic alignment, I clicked on the *Start import* button to import the matched segments as translation units to the project TM. A problem was encountered with the results. Since there is no option to merge segments during manual alignment, I aligned them in an n:1 relation, which resulted in two SL segments having the same TL segment matched to them:

| | | | |
|---|---|---|---|
| SL1 | Ich wollte dir nur sagen, dass ich | TL1 | Ich will nicht mit dir reden. |
| SL2 | nicht mit dir reden will, okay? | TL1 | Ich will nicht mit dir reden. |

If there were settings available for alignment where I could adjust the segmentation rules to 'ignore line breaks for segmentation' or if there was the option to merge segments during manual alignment, this problem could easily be resolved. Since both Across and SDL Trados lack these options, I had to manually correct the segments during the TMX export, which thankfully *is* possible in Across.

This tool is, nevertheless, mentioned in this section as one of the useful components of a CAT tool because it can be useful to e.g. align translated book sequences to translated film sequences in cases such as *Lord of the Rings, Harry Potter, Game of Thrones*, etc. This may only be possible when the user builds up huge corpora of transcripts and eBook contents, but in the case of a subtitle-providing company, this tool could be used to create a TM for reference. This TM could then be distributed to individual subtitlers for use either in the cloud or for download from an FTP server.

With regard to the outcome of this field test, one big learning identified was that using different tools to compile their outputs in one tool is easier said than done. That constitutes yet another argument in favour of creating a tool that handles translation and subtitling in one go. Now, the core function of a CAT tool will be tested: the translation memory.

### 4.3.2. Field test (5): TM of Transit NXT and SDL Trados Studio 2014

To test a CAT tool that is capable of translating subtitles, I installed Transit NXT by STAR on my laptop – for being the only CAT tool offering multimedia synchronisation during translation – and selected the user role *Translator* out of *Project Manager, Project Calculation Manager, Translator, Reviewer, Markup Specialist, Reference Material Manager, Alignment Specialist, Localisation Specialist, Terminology Manager, Terminologist, Terminology Translator, Super User.* Here I could create a *New project from scratch.* I chose English as SL and German as TL. For source file type, there is three subtitling file types available: SubRip, Text, WebVTT. I selected SubRip and selected my file for translation. No reference projects are available. The minimum quality for fuzzy matches and fuzzy statistics is set to 70%. Once I opened the file in editing mode, I could choose MT, but would need an API key for Google Translate which I do not have.

After struggling with the application to find the basic settings such as how to include a TMX for fuzzy matches and due to the unintuitive file storage in the folder (as mentioned above in Sub-section 3.3.2), I decided to test the TM function within SDL Trados instead, since testing the import, usage, and export of a TM is the core aim of this chapter. I only wanted to know how many repetitions would be in the recaps and how efficient the TM will be. Therefore, no video playing is necessary for this test. SDL Trados, furthermore, offers a user-friendly creation and handling of files, which are all stored in one project folder and can be opened one by one directly from a folder with no need for the user to run the software first.

I built a TM of 74 entries by aligning two recap scenes of previous episodes using Across because SDL Trados does not allow for an easy manual review of the aligned segments. The new recap scene that I opened for translation had 111 words, of which the TM pre-translated 29 per cent using fuzzy matches, leaving me with 78 words to translate. Those 29 per cent were represented by four segments (out of 15) with the fuzzy results consisting of one context match, one 83% match and two 100% matches.

Whether the creation of such a TM, on the downside, is worth the effort is questionable. It took me hours to transcribe the recap scenes word by word including time codes, to then edit them down to shorter subtitles, save both versions in a plain text file and edit it again so the alignment tool does not segment by line breaks, to then align the segments correctly and export a usable TMX, just to import the TMX and finally use it to get to a result of four matches or almost a third of a recap scene. Then again, since dialogue in sitcoms and drama series is usually not the most philosophical and complicated, a TM comprising of more than 74 entries could provide even more fuzzy matches, given that the translation units are stored accurately.

Such a TM could be used by a translator community or a translation agency as a master TM that can be accessed online via VPN, FTP, or a simple cloud account and used by the translator for the translation of e.g. a series. To make sure for this master TM to be of any use, a quality manager is needed. The quality manager will have to run quality checks in the TM on a regular basis to eliminate multiple TL segments for one SL segment, or if several matches shall be kept, to tag them appropriately with context and approve that the translations are correct. So, what does quality assurance in translation entail?

### 4.3.3. Quality Assurance (QA)

Regardless of the way of building-up a TM, before and after alignment and translation, QA needs to be done to 1) compare and assure quality in the exported TM, but then 2) compare the newly proposed segments to the existing master TM segments as well. There are integrated QA tools with extensive adjustment options in SDL Trados and Transit NXT. For an even more sophisticated and in-depth QA, plugins are available for download. They are usually payable options, such as Xbench, which in one step scans through many documents, spots inconsistent translation or more than one translation for a SL segment, and points out deviations from mandatory use of certain terms from the term base.

QA is a mandatory step after both translation and subtitling. While for translation, the CAT tool has extended functionalities, such as comparison of tags and punctuation, identification of empty TL segments, mandatory use of terms and translation unit consistency checks, etc., a subtitling software's linguistic QA is limited to spelling, but adds technical factors like line length, wpm rate, and gap restrictions, which the CAT tool, again, lacks. I therefore advise to merge the quality managing functionalities of both translation and subtitling tools to gain a powerful tool that supports checking of all attributes of a subtitle and tackles translation issues at the same time.

A subtitle's semantic dimension adds to the challenging task of subtitling, as identified in 3.1.3 (d). Therefore, another check could be added: semantic unit checks. At the end of subtitling, the file should be examined to identify articles and prepositions. If the software identifies some, and the preset values of line length and number of lines can be observed, the software should insert a non-breaking space between article and noun or preposition and the subsequent word. This rule also applies to punctuation which should be followed by an automatic line break in the given space constraints. The checks of a TM should be added to the subtitling software's usual QA features to build a powerful translation tool for subtitling environments.

### 4.3.4. Conclusion: TM in AVTEnT

Although the literal transcript may be already cued and stored in the SRT, I should pay attention to the fact that it might take some effort to edit down the SDH according to the new subtitle's limitations resulting from the need of sound descriptions. The same goes for interlingual subtitling: A TM bears the difficulty that the TL's prerequisites are different than the SL's conditions, such as reading speed and maximum number of lines per subtitle. A solution to this may be to have two columns with the SL on the left and the TL on the right and their respective settings applied separately. The TM for both intralingual and interlingual subtitles should propose longer translation units as well to give a greater choice to the translator. So, if there is a subtitle that adheres to one guideline or style guide, it might not do so for another customer.

When a subtitle file is newly created, without a transcript (auto-transcription) at hand, as for SDH, the file should be auto-transcribed and post-edited if needed, to be stored as SL segments. It is vital that either the translator/subtitler or a quality manager defines the parameters for the subtitles, so that the software can automatically assign fuzzy matches to the positions in the video and highlight units exceeding wpm rates, line and characters-per-line restrictions. Furthermore, context should be included in the fuzzy match display, so the translator knows what genre and mood the fuzzy match is coming from. Context for mood could be e.g. the stored subtitle’s preceding and subsequent lines of dialogue.

There could also be translation units for theme songs of e.g. children’s shows which the translator can store in a show template as a reference file, so the software inserts the lyrics and other information automatically and time-codes the episodes consistently throughout the show. If there is no match in the TM, MT can be of help. The translation layout should be side-by-side to avoid extensive eye-movement. Saving files should be easily maintainable by using straight forward structures: The user can store files in a project and choose from a range of project types that the user would like to open: monolingual source, monolingual target, or a merge to a bilingual product.

The TM could be used either in a cloud or as an offline application. The option to export or import a TM must be ensured. What if a too long subtitle is stored in a TM and does not comply with the guidelines of another company? These reference files and the fact that many companies use different style guides lead me to think of setting up templates to save time, observe quality, and thus, save costs for translators. This item will be discussed in *4.5 Cloud-based CAT tool and customer templates*. Next issue to be discussed, however, is the use of term bases and thesauri in translation technology that could be helpful in subtitling environments as well.

## 4.4. Term base (TB), web search, and thesaurus

Term databases can be built-up or downloaded from the internet by the translator for different domains, projects, or customers. They scan through the SL text and compare terms or groups of words in the SL against stored items in the TB. When there is a match, the TB highlights the SL terminology, and once the translator types in the first letter of the translation in the TL column, the TB auto-suggests the full translation. Since a TB is a useful compound of CAT tools, I consider it indispensable for an AVT environment. While TBs serve as glossaries of specialised or general terminology, a thesaurus, on the other hand, serves as a dictionary of synonyms. An integrated web search would make the whole terminology management even easier.

### 4.4.1. Term base (TB)

Within CAT tools, term bases (TB) serve the purpose of a glossary search for specialised terms. Term bases are similar to e(lectronic)-dictionaries, although they differ from e-dictionaries in that they contain vocabulary of specialised areas and are often generated by international institutions for public use, such as IATE[24], the EU's online term base. Usage of term bases is advisable, even if the language is not for specialised purposes. In a source text, active terminology recognition can be activated, so the term base identifies and highlights a SL term to then look up and propose the TL term to the translator automatically. The TB, thus, facilitates a lookup of words, although a manual search by subject field, synonym, definition etc. is possible with exact and fuzzy results being offered. Some TB applications allow for a wild card seach using such symbols as *, %, or $. When a word of search is spelled incorrectly, the application is able to recognise it and delivers fuzzy matches. Also, a proximity search by using quotation marks can lead to a more precise result (Kenny 2011: 457-460).

What exactly is the difference between a TM and a TMS then? A TM saves segments of entire sentences or paragraphs and can retrieve 100% corresponding matches or variations of this segment, called fuzzy matches. By using the concordance search in a TM, a group of words can be found in a segment. In contrast to looking up sentences or paragraphs, a term base looks up particular terms or groups of words that are stored in a glossary. The term base identifies SL entries and auto-suggests a TL entry from the glossary to the user if there is a full or fuzzy match. The term can also be looked up manually if it is not recognised by the tool itself. So, if there is a translation unit that contains the word in need, but the fuzzy match sensitivity threshold is set to a high value (the minimum threshold is 30% similarity), the TM

---

[24] For more see http://iate.europa.eu/

would not suggest it as a fuzzy match, but the term base would get a result for the terminology:

| | |
|---|---|
| You know, you can say it backwards, which is ‘docious-ali-expi-istic-fragil-cali-rupus’ - but that's going a bit too far, don't you think? | Das Wort kann man auch rückwärts sagen:<br>Getischaliexpilisticfragicalisuper,<br>aber das geht zu weit, findest du nicht auch? |
| docious-ali-expi-istic-fragil-cali-rupus | [no match as the fuzzy match threshold would need to be lower than 30%] |

*Figure 19: Example of no match for individual words within TM (source: Walt Disney’s* Mary Poppins, *1964)*

TBs and their functionality are powerful tools and are a mandatory part of a CAT tool for consistency throughout many projects. TMS managers can store different approved terms, for instance, for different e-learning providers with the respective ‘do not use terminology’ or mandatory terminology for certain customers. An activated automatic term recognition of *Eye on Springfield* can therefore have the translator refrain from using different translations or from translating a passage at all (e.g. in one scene of *The Simpsons*, Homer sings in German a translated version of Luka by Suzanne Vega ‘Mein Name ist Luka, ich lebe im zweiten Stock’ for which there is no obvious reason) and thus promote consistency.

In *3.1 Forms of subtitling and their technological needs*, a TB was identified in (b) to look-up intro and theme songs in a template (e.g., stored in the application), spot and transcribe them in a unified way throughout episodes, which I need to revise now that the TM has been identified as the primary solution for this issue. For the handling of dialectal expressions and taboo words (as identified in 3.1.1 (h)), and customer-specific as well as series-specific terms and phrases (see 3.1.2 (d)), I proposed a TB and now consider it useful for the named instances.

In both intralingual and interlingual subtitling of series and repetitive content of, e.g., e-learning courses, the customer or agency can manage terminology, save it in a reference file and enable access for the translator. The subtitler could add and store a customised list of frequently used terms in the TB which it then auto-recognises. If it is a cloud TB, the customer or agency would need to approve the suggestions made by the translator. For intralingual subtitling, a term base should be available for ASR to a) identify dialects and b) tag them; c) match dialectal expressions with standard variation and have translator choose between standard and dialect as per customer needs; d) store the samples for faster speech recognition in the future. Another useful feature during translation is, of course, using the internet for research and lookups of terminology.

### 4.4.2. Web search

SDL Community Developers developed the free plugin app *Web Lookup!* that allows users to perform an integrated web search from within the CAT tool environment, thus enhance their workflow: they just need to mark the SL or TL word they would like to look up and either right-click and select the web lookup or press *Alt+F9* to perform a search. An integrated window opens the preferred website automatically, such as Linguee.com, dict.cc, Merriam-Webster, Oxford Learner's Dictionaries, Duden, or simply Wikipedia. Users can customise the list of search engines that they would like to consult during translation and therefore do not need to leave the tool and switch to the web browser, making the internet search faster, more efficient, and external research obsolete. This plugin would also enhance the workflow during subtitling and could as well be used for websites providing thesauri.

### 4.4.3. Thesaurus

Díaz Cintas proposes to 'include thesauri and suggest synonyms when space restrictions are at a premium' (2015: 641). A thesaurus is a dictionary of same-language synonyms. This would be particularly helpful in subtitling, as subtitles often need to retain indispensable elements while omitting the dispensable ones and condensing the partly dispensable ones (Kovačič 1991: 409), and at the same time, use shorter words and condensed speech. In combination with a TM or ASR, the thesaurus could be helpful insofar as it could analyse fuzzy matches, identify terms and their length and suggest shorter synonyms that fit into the space and time constraints of the subtitle. The tool should also identify frequently used terms and auto-suggest shorter synonyms for use.

### 4.4.4. Proposed process

When a subtitle file is newly created, with or without a transcript (speech-to-text alignment vs. auto-transcription) to hand, the video file will be scanned by the software to transcribe intralingual files and suggest MT and TM units for interlingual files. In the process, the term base a) identifies customer-specific and TV show-specific terminology to highlight it and suggest translations; b) identifies dialectal words and phrases and suggests—according to the customer's style guide—to either use standard language or suggests spelling. Furthermore, the TB helps the ASR to recognise dialectal words and phrases and train it. For episodes from a series, the software inserts the theme song in a unified way for all episodes (same duration, same gaps, same number of lines in one subtitle).

Subtitles that are too long are identified and highlighted by the software. The thesaurus a) identifies long words to b) show shorter alternatives with their number of characters besides them; c) identifies interjections and appellative names to d) suggest to omit them. To facilitate web search without leaving the application, an integrated customisable web search tool is available with a keyboard shortcut assigned to it and the possibility to easily copy and paste results into the subtitling application.

## 4.5. Cloud-based CAT tool and customer templates

A customer template was identified for such attributes as theme songs (3.1.1. (b)) and on how the software is to handle music, lyrics, and dialectal (h) or foreign language utterances (g) whether it should auto-colour or auto-tag identified speakers, and how to handle speech that crosses shot changes (3.1.2. (c)) among other rules (3.1.2. (e), 3.1.3. (a), (b)). Below, an example of an exported WinCAPS customer template is presented. In addition to all these attributes, the user should be able to store such information as of how to handle dialects and taboo words, how to denote music and whether to colour-code speakers or tag them. In addition to that, the user should be able to save a show-related reference file containing character names and theme songs.

```
AllowOverlaps = FALSE
ColourBack = Black
ColourFore = White
ColourLocked = FALSE
ContinuationEnabled = FALSE
DotsAfterComma = FALSE
DotsOnSecondSubtitle = TRUE
EnableBlankSubtitles = FALSE
FileLocked = FALSE
FontBold = FALSE
FontItalic = FALSE
FontName = Arial
FontSize = 30
FontUnderline = FALSE
HorizontalPosition = 0.500000
Justify = 0.500000
Language = German
LockOnClose = FALSE
MaximumDuration = 200
MaximumRowCount = 2
MaximumRowLength = 36
MinimumDuration = 25
MinimumGap = 1
Personality = Teletext
```

*Figure 20: Wincaps customer template configurations export*

Many companies use different style guides. These style guides and the show-specific reference files (characters, speaker colours, theme songs) could be saved in a template that can be downloaded by the translator to save time, observe quality, and thus, save costs for translators. This will require the customer to constantly update the style guide with various parameters for quality assurance.

GeoWorkz’s Translation Workspace (TWS) is a cloud-based translation system with a monthly plan. Companies benefit from this tool in many ways: It features project management and quality modules. Furthermore, translators have to pay their monthly fee based on a maximum word count that will be translated per month, and at the same time they are required to use the tool. It is used and also offered by Lionbridge. In other words, if translators want to work for Lionbridge, they must pay for usage of their tool and have no alternative as Lionbridge requires them to use TWS. In my humble opinion, all these tools are far away from being user-friendly, but I like the idea of a cloud-based solution for translators. This is also what Memsource had in mind when they started their cooperation with hakromedia GmbH (as mentioned in Sub-section 3.3.3).

First of all, the translation units are stored in the cloud as the translator moves along, so there is no need to worry about losing data, e.g. when the laptop battery dies or the translator forgets to save the file before shutting down the computer. Then again, a stable internet connection is needed at all times, otherwise the translation progress cannot be saved in the cloud. Concordance search and term base search are both possible. Usability and user-friendliness depends on the individual solution, and so do keyboard shortcuts. Whether or not there are any available varies from solution to solution. What all tools have in common is that the project management including assigning, forwarding and unlocking access to different TMs is usually carried out by the PM of a project. The fact that TMs are stored in many various bidirectional language pairs is a huge advantage for translators. That way, translators can also use TMs of other languages if they are stuck on a segment and need inspiration for translation possibilities. The tool, however, could as well be developed as a subtitling plugin for CAT tools or CAT tool add-in for subtitling programs. Being one of the first such AVTEnTs, anything that supports the translator will work.

# 5. Conclusions

Internet use is growing around the globe, with video platforms generating more data on a daily basis than humanity did in total in the whole of history until 2000 (see more on history in Section 2.3), and thus creating the need for more subtitles than ever. Yet technologies for translators in subtitling are not keeping pace and still retain their established separation into two major areas of tool suites: CAT tools for translators on the one hand and subtitling software for subtitlers on the other.

Combining the two suites has been a much-requested feature from academia and practitioners alike over the past decades: Carroll states that ‘[a]ll types of subtitling would benefit from enhanced software solutions, ranging from integrated CAT (computer-assisted translation) tools to automatic voice and cut recognition, which are starting to appear” (2004:

3), the SUMAT project tried to make MT useful in subtitling, and Díaz Cintas has found that ‘software engineers do not seem to have made any serious attempts to develop tools […] that would help subtitlers with the linguistic dimension and not only with the technical tasks’ (2015: 641). The aim of this thesis, therefore, was to propose a solution that supports translators who work in subtitling so they can carry out translation, subtitling, and post-editing tasks all in one tool. The main goal of this supporting technology is both to raise quality and productivity for the translator and to ensure consistency and accuracy for the customer.

To achieve this aim, following an introduction to translation and sutbtitling technology presented in Section 1, Section 2 highlights the history of subtitle translation, raises the question of productivity and quality in subtitling, and discusses arguments in favour of a CAT tool to be used in subtitling environments.  From existing guidelines, laws, and customer-specific style guides I conclude that subtitling is a professional field of translation important enough to justify investment in technology research and development for it. Yet, Section 3 shows that little development in this field has taken place to date. My investigation of the market led to only one very expensive platform that provides translation features for subtitling processes: a collaboration between the cloud-based CAT tool Memsource and the multimedia translation provider Hakromedia, with the product being subCloud, a cloud-based subtitling tool with translation technology elements. After analysing the different types of subtitling and the issues that they raise, I identified a range of solutions, i.e. functionalities that a subtitle translation tool, or AVTEnT (audiovisual translation environment tool) as I call it, should include.

As regards *intralingual* subtitling, especially for d/Deaf and hard-of-hearing people, the AVTEnT should include an ASR to auto-transcribe and set time codes automatically while observing such settings as minimum gaps, line length, reading speed, scene cuts, theme songs and lyrics, etc. Music identification tools should help translators locate artists, their songs, and lyrics so that the translators do not need to exit the application. If a TM is used, the SL should be represented by the word-by-word transcription, while TL is the stored post-edited segments. A very necessary tool is the automated colouring or tagging of speakers, for which the ASR should be trained by automatically storing vocal samples of actors and dubbers of a TV show. If possible, the auto-transcription should include sound descriptions.

For *interlingual* subtitling, the tool should auto-align speech to text, when the translator feeds a transcript into the translation feature consisting of a) a translation memory with translation units that include subtitle information like line length and wpm rates together with context of setting and show title; b) a term base to automatically suggest show- and customer-specific, dialectal, and other terminology to ensure consistency throughout periodically repeated shows or e-learning courses; c) an integrated web search so the translator does not need to switch applications during the subtitling process; d) machine translation features to suggest fuzzy matches from which the translator can choose when post-editing; and e) quality assurance elements that merge the QA elements of CAT tools with the technological checks contained in subtitling software, such as wpm rates, overlaps, and line limitations, but should also include semantic unit checks. The findings of previous studies show that the use of a TM combined with MT leads to higher productivity and better-quality outputs.

For *all forms of subtitles*, a customer template should be available with parameters that can be linked to the term base and TM to automatically apply style guide rules. One element that it would, in my view, be beneficial to have but that is not absolutely mandatory as part of my proposal is an optical character recognition (OCR) system that automatically identifies open subtitles, such as integrated subtitles or intertitles so as to not cover them with closed subtitles; the software should therefore block the space so the subtitler cannot create a subtitle for a certain period of time, usually one second.

When I look at my proposed solutions, a very important question comes to mind: Do all these features downgrade translator-subtitlers to mere post-editors because the full automation of processes makes subtitling a pure post-editing task? Since this is a tool to support translators in their workflow and thereby raise productivity while at the same time assuring quality, my view is that it does not in the least detract from, but actually adds to professionalism. I cannot see how either linguistic editing down to condense mood- and plot-conveying language while retaining mood and plot elements, or the purposeful application of technology with target audience-based decision-making, or the use of quality-assuring features could in any way deprofessionalise subtitling as an occupation; rather, they all make the workflow more efficient and help subtitlers save time and money.

### ***Further research***

One possible avenue for further research would be to survey how keen subtitlers would be to use such a tool, which could be included in broader research on the usability of the tool proposed in this thesis. The focal questions of such a user survey should be: Are the CAT tool features proposed for subtitling programs useful in intralingual and interlingual translation processes? Does the tool enhance productivity and quality? Is it user-friendly? Does it result in a shorter turnaround time than that of translators who create subtitles from scratch? I therefore propose to create a written survey and to send it to a group of around 30–50 professional subtitlers. For a more solution-oriented approach, I recommend presenting the problem and having the respondents choose from a series of possible solutions by asking which option within the software they would prefer to work with. Based on conclusions from such a survey, a powerful tool can be developed that enhances the process in subtitle translation in its entirety.

Other than that, it is regrettable that academia and also practitioners have not provided more evidence in favour of such a CAT tool to be used in subtitling and more articles refuting the arguments against such a tool. My opinion remains unchanged: such a tool presents an opportunity to use technology to assist translators in their daily work. Those who are not happy to use it are perfectly entitled to stick to their current practice. Whether to use the AVTEnT in a cloud or as a plug-in in offline applications can be discussed in further usability studies on this topic. The usefulness of an AVTEnT in such crowd-translated contexts as Netflix and YouTube, however, is, I believe, indisputable.

# References

ADLAB - Audio Description: Lifelong Access for the Blind. (2014). *Final Report - Public Part.* Education, Audiovisual & Culture Executive Agency. Retrieved June 28, 2017, from http://www.adlabproject.eu/Docs/ADLAB%20report%20public.pdf

ALPAC (Automatic Language Processing Advisory Committee). (1966). *Languages and Machines: Computers in Translation and Linguistics.* A Report by the Automatic Language Processing Advisory Committee, Division of Behavioral Sciences, National Academy of Sciences, National Research Council, Washington, DC: National Academy of Sciences, National Research Council, 1966.

*AMS (Audiovisual Media Services) Directive Summary.* (last updated 2015, September 21). Retrieved January 20, 2017, from EUR-Lex: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=legissum:am0005

BBC R&D - Sandford, J. of BBC. (last updated 2016, March 4). *How fast should subtitles be?* Retrieved June 12, 2017, from BBC Research & Development Blog: http://www.bbc.co.uk/rd/blog/2015-09-how-fast-should-subtitles-be

Bowker, L. (2002). *Computer-Aided Translation Technology: A Practical Introduction.* Ottawa: University of Ottawa Press.

Bowker, L. (2005, March). Productivity vs Quality? A pilot study on the impact of translation memory systems. *Localisation Focus - The International Journal of Localisation, 4*, pp. 13-20. Retrieved from https://www.localisation.ie/sites/default/files/publications/Vol4_1Bowker.pdf

Bowker, L., & Fisher, D. (2010). Computer-aided translation. In Y. Gambier, & L. v. Doorslaer, *Handbook of Translation Studies 1* (pp. 60-65). Amsterdam/Philadelphia: John Benjamins Publishing Company.

CAB ACR - Canadian Association of Broadcasters. (2016, March). *CAB Update - Volume 6.* Retrieved July 2, 2017, from http://www.cab-acr.ca/english/updates/update_vol6.pdf

Carroll, L. (1872). *Through the looking glass, and what Alice found there.* London: Macmillan & Co.

Carroll, L. (1974). *Alice hinter den Spiegeln.* (C. Enzensberger, Trans.) Berlin: Macmillan & Co.

Carroll, M. (2004). Subtitling: Changing Standards for New Media. *The LISA Newsletter: Globalization Insider XIII/3.3*.

Chan, S.-w. (2015). The development of translation technology. In S.-w. Chan, *The Routledge encyclopedia of translation technology / edited by Chan Sin-wai* (pp. 3-31). Oxon, New York: Routledge.

Chaume, F. (2013). The turn of audiovisual translation. In *Translation Spaces 2* (pp. 105-123). Amsterdam: John Benjamins Publishing Company. doi:10.1075/ts.2.06cha

CORDIS (Community Research and Development Information Service). (2014, March 5). *CIP - Competitiveness and innovation framework programme (CIP) (2007-2013)*. Retrieved June 16, 2017, from European Commission: http://cordis.europa.eu/programme/rcn/838_en.html

Declercq, C. (2015). Editing in translation technology. In S.-w. Chan, *The Routledge encyclopedia of translation technology / edited by Chan Sin-wai* (pp. 480-493). Oxon, New York: Routledge.

Díaz Cintas, J. (2015). Technological Strides in Subtitling. In S.-w. Chan, *The Routledge encyclopedia of translation technology / edited by Chan Sin-wai* (pp. 632-643). Oxon, New York: Routledge.

Díaz Cintas, J., & Anderman, G. (2009). *Audiovisual Translation. Language Transfer on Screen.* Basingstoke: Palgrave Macmillan.

Díaz Cintas, J., & Anderman, G. (2009). Introduction. In J. Díaz Cintas, & G. Anderman, *Audiovisual Translation. Language Transfer on Screen* (pp. 1-17). Basingstoke: Palgrave Macmillan.

Díaz Cintas, J., & Orero, P. (2010). Voiceover and dubbing. In Y. Gambier, & L. v. Doorslaer, *Handbook of Translation Studies 1* (pp. 441-445). Amsterdam/Philadelphia: John Benjamins Publishing Company.

*DIRECTIVE 2010/13/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL.* (2010, March 10). Retrieved January 20, 2017, from EUR-Lex: http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32010L0013&rid=1

Etchegoyhen, T., Bywood, L., Fishel, M., Georgakopoulou, P., Jiang, J., van Loenhout, G., . . . Volk, M. (2014). Machine Translation for Subtitling: A Large-Scale Evaluation. *LREC 2014, Proceedings of the Ninth International Conference on Language*

*Resources and Evaluation* (pp. 46-53). Reykjavik: ELRA - European Language Resources Association. Retrieved June 16, 2017, from http://www.lrec-conf.org/proceedings/lrec2014/pdf/463_Paper.pdf

European Broadcasting Union (EBU). (1991, February). *Specification of the EBU Subtitling data exchange format.* Retrieved from EBU Operating Eurovision and Euroradio: https://tech.ebu.ch/docs/tech/tech3264.pdf

Fox, W. (2012). *Integrierte Bildtitel - Eine Alternative zur traditionellen Untertitelung. Am Beispiel der BBC-Serie Being Human.* Master's Thesis at Johannes Gutenberg University Mainz/Germersheim.

Gambier, Y., & Doorslaer, L. v. (2010). *Handbook of Translation Studies 1.* Amsterdam/Philadelphia: John Benjamins Publishing Company.

Georgakopoulou, P. (2006, July). Subtitling and Globalisation. *JosTrans*(Issue 06). Retrieved June 12, 2017, from http://www.jostrans.org/issue06/art_georgakopoulou.php

Göpferich, S. (2010). Transfer and Transfer Studies. In Y. Gambier, & L. v. Doorslaer, *Handbook of Translation Studies 1* (pp. 374-377). Amsterdam/Philadelphia: John Benjamins Publishing Company.

Gottlieb, H. (2002). Untertitel: Das Visualisieren filmischen Dialogs. In H.-E. Friedrich, *Schrift und Bild im Film* (pp. 185-214). Bielefeld: Aisthesis-Verlag.

Guerberof Arenas, A. (2008). *Productivity and quality in the post-editing of outputs from translation memories and machine translation.* Universitat Rovira i Virgili: PhD Dissertation.

Guerberof Arenas, A. (2014, December). Correlations between productivity and quality when post-editing in a professional context. *Machine Translation, 28*(3-4), pp. 165-186.

Harrenstien, K. (2009, November 19). *Automatic captions in YouTube.* Retrieved June 28, 2017, from Google Official Blog: https://googleblog.blogspot.co.at/2009/11/automatic-captions-in-youtube.html

Hillman, R. (2011). Spoken word to written text: Subtitling. In K. Malmkjaer, & K. Windle, *The Oxford handbook of translation studies / ed. by Kirsten Malmkjaer & Kevin Windle* (pp. 379-393). Oxford [et al.]: Oxford University Press.

IDC - International Data Corporation. (2014). *The Digital Universe of Opportunities.* sponsored by Dell EMC. Retrieved June 11, 2017, from https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

Ivarsson, J. (last updated 2003, December 12). *Transedit*. Retrieved November 29, 2016, from Glossary of subtitling terminology: http://www.transedit.se/glossary.htm

Ivarsson, J. (last updated 2004, November 17). *Transedit*. Retrieved November 29, 2016, from A SHORT TECHNICAL HISTORY OF SUBTITLES IN EUROPE: http://www.transedit.se/history.htm

Ivarsson, J., & Carroll, M. (1998). *Subtitling.* Simrishamn: TransEdit HB.

Jüngst, H. E. (2010). *Audiovisuelles Übersetzen: Ein Lehr- und Arbeitsbuch.* Tübingen: Narr Francke Attempto Verlag.

Karamitroglou, F. (1998, April 2). A Proposed Set of Subtitling Standards in Europe. *Translation Journal, 2*(2). Retrieved June 30, 2017, from http://translationjournal.net/journal/04stndrd.htm

Kenny, D. (2011). Electronic Tools and Resources for Translators. In K. Malmkjaer, & K. Windle, *The Oxford handbook of translation studies / ed. by Kirsten Malmkjaer & Kevin Windle* (pp. 455-474). Oxford [et al.]: Oxford University Press.

Khoso, M. (2016, May 13). *How Much Data is Produced Every Day*. Retrieved March 13, 2017, from Northeastern University: http://www.northeastern.edu/levelblog/2016/05/13/how-much-data-produced-every-day/

Koponen, M. (2016, January). Is machine translation post-editing worth the effort? A survey of research into post-editing and effort. *The Journal of Specialised Translation*(25), pp. 131-148. Retrieved June 20, 2017, from http://www.jostrans.org/issue25/art_koponen.pdf

Koponen, M. (2016). *Machine Translation Post-editing and Effort: Empirical Studies on the Post-editing Process.* University of Helsinki: Dissertation.

Kovačič, I. (1991). Subtitling and contemporary linguistic theories. In M. Jovanović, *Translation, A Creative Profession: Proceedings XIIth FIT World Congress (Belgrade, 2-9 August 1990)* (pp. 407-417). Belgrade: Prevodilac.

Leeson, L., & Vermeerbergen, M. (2010). Sign language interpreting and translating. In Y. Gambier, & L. v. Doorslaer, *Handbook of Translation Studies 1* (pp. 324-328). Amsterdam/Philadelphia: John Benjamins Publishing Company.

Linde, Z. d., & Kay, N. (1999). *The Semiotics of Subtiling.* Manchester: St. Jerome Publishing.

LISA - Localization Industry Standards Association. (2001, August 29). *TMX Format: Specifications.* Retrieved May 29, 2017, from Translation, Training/education, and Testing: http://www.ttt.org/oscarstandards/tmx/tmx13.htm

Luyken, G.-M. (1991). *Overcoming language barriers in television: dubbing and subtitling for the European audience.* Manchester: European Institue for the Media.

Malmkjaer, K., & Windle, K. (2011). *The Oxford handbook of translation studies / ed. by Kirsten Malmkjaer & Kevin Windle.* Oxford [et al.]: Oxford University Press.

Memsource. (2016, February 11). *HAKROMEDIA SUBCLOUD CONNECTOR TO MEMSOURCE*. Retrieved June 30, 2017, from https://www.memsource.com/blog/2016/02/11/hakromedia-subcloud-connector/

Moffat, S., & Gatiss, M. (Directors). (2010-). *Sherlock* [Motion Picture].

Netflix. (2017). *You Me Her*. Retrieved July 1, 2017, from Netflix Media Center: https://media.netflix.com/en/only-on-netflix/116016

Ofcom. (2014, November 5). *Delays in TV subtitling remain an issue for viewers*. Retrieved June 12, 2017, from Ofcom media releases: https://www.ofcom.org.uk/about-ofcom/latest/media/media-releases/2014/subtitling-second-report

Ofcom. (2015, November 27). *Measuring live subtitling quality: Results from the fourth sampling exercise.* Retrieved January 20, 2017, from Ofcom: https://www.ofcom.org.uk/__data/assets/pdf_file/0011/41114/qos_4th_report.pdf

Ofcom. (2015, May 13). *Ofcom's Code on Television Access Services.* Retrieved January 20, 2017, from https://www.ofcom.org.uk/__data/assets/pdf_file/0016/40273/tv-access-services-2015.pdf

Ofcom. (2017, January 30). *Ofcom's Code on Television Access Services.* Retrieved June 29, 2017, from https://www.ofcom.org.uk/__data/assets/pdf_file/0020/97040/Access-service-code-Jan-2017.pdf

Ofcom. (n.d.). *What is Ofcom?* Retrieved June 27, 2017, from Ofcom: https://www.ofcom.org.uk/about-ofcom/what-is-ofcom

Orero, P. (2007). Sampling audio description in Europe. In J. Díaz Cintas, P. Orero, & A. Remael, *Media for All: Subtitling for the Deaf, Audio Description, and Sign Language* (pp. 111-126). Amsterdam, New York: Rodopi.

ORF. (n.d.). *Hören statt sehen: Fernsehen für Blinde und Sehbehinderte*. Retrieved January 3, 2017, from ORF Audiodeskription: http://der.orf.at/kundendienst/service/audiodeskription104.html

Pöchhacker, F. (2004). *Introducing interpreting studies.* London, New York: Routledge.

Popper, R. (Writer), & Bendelack, S. (Director). (2011–). *Friday Night Dinner, S01E05* [Motion Picture].

Remael, A. (2010). Audiovisual translation. In Y. Gambier, & L. v. Doorslaer, *Handbook of Translation Studies 1* (pp. 12-17). Amsterdam/Philadelphia: John Benjamins Publishing Company.

Romero-Fresco, P., & Martínez Pérez, J. (2015). Accuracy Rate in Live Subtitling: The NER Model. In R. Baños Piñero, & J. Díaz Cintas, *Audiovisual Translation in a Global Context: Mapping an Ever-changing Landscape* (pp. 28-50). Palgrave Macmillan UK.

Schwarz, B. (2011). Translation for dubbing and voice-over. In K. Malmkjaer, & K. Windle, *The Oxford handbook of translation studies / ed. by Kirsten Malmkjaer & Kevin Windle* (pp. 394-409). Oxford [et al.]: Oxford University Press.

Screen Subtitling Systems Ltd. (n.d.). *Wincaps Q4 Brochure.* Retrieved from Screen Systems: http://subtitling.com

SDL Trados. (n.d.). *History*. Retrieved from SDL Trados: http://www.sdltrados.com/resources/infographics/history-timeline.html

SDL Trados Studio 2017. (n.d.). *AdaptiveMT engines adapt to your style, terminology and content.* Retrieved July 16, 2017, from SDL Trados Help: https://docs.sdl.com/LiveContent/web/pub.xql?action=home&pub=SDL%20Trados%20Studio%20Help-v4&lang=en-US&docid=GUID-A4DDDFFF-3FD2-4F25-ADBA-B15EF51CC4E5&query=%2B%22adaptive%22

Stevenson, R. (Director). (1964). *Mary Poppins* [Motion Picture].

Taylor, S., Theobald, B.-J., & Matthews, I. (2015, April 19). *A Mouth Full of Words: Visually Consistent Aucoustic Redubbing.* Retrieved June 17, 2017, from Disney Research: https://www.disneyresearch.com/publication/a-mouth-full-of-words-visually-consistent-acoustic-redubbing/

The Guardian. (2013, May 17). Ofcom to crack down on sloppy subtitles. Retrieved May 30, 2017, from https://www.theguardian.com/media/2013/may/17/ofcom-crack-down-poor-subtitles

The Guardian. (2015, January 28). YouTube staff too swamped to filter out all terror-related content. Retrieved March 30, 2016, from https://www.theguardian.com/technology/2015/jan/28/youtube-too-swamped-to-filter-terror-content

ÚFAL - Institute of Formal and Applied Linguistics. (n.d.). *CzEng 1.6*. Retrieved July 4, 2017, from UFAL: http://ufal.mff.cuni.cz/czeng

Wang, N. (2017, March 23). *Visualizing Sound Effects.* Retrieved May 15, 2017, from YouTube Engineering and Developers Blog: https://youtube-eng.googleblog.com/2017/03/visualizing-sound-effects.html

Wilss, W. (1994). Grundkonzepte der Maschinellen Übersetzung. In I. Fischer, K.-H. Freigang, F. Mayer, & U. Reinke, *Sprachdachtenverarbeitung für Übersetzer und Dolmetscher: Akten des Symposiums zum Abschluß des Saarbrücker Modellversuchs, 28./29. September 1992* (pp. 169-188). Hildesheim, Zürich, New York: Georg Olms Verlag.

YouTube. (2010, March 4). The Future Will Be Captioned: Improving Accessibility on YouTube. Retrieved June 28, 2017, from https://youtube.googleblog.com/2010/03/future-will-be-captioned-improving.html

YouTube. (2017). Global Reach. Retrieved December 15, 2016, from https://www.youtube.com/intl/en-GB/yt/about/press/

YouTube. (n.d.). *Supported subtitle and closed caption files*. Retrieved June 16, 2017, from YouTube Help: https://support.google.com/youtube/answer/2734698?hl=en&ref_topic=7296214

YouTube. (n.d.). *Use automatic captioning*. Retrieved June 16, 2017, from YouTube Help: https://support.google.com/youtube/answer/6373554?hl=en

ZDF. (n.d.). *Untertitel-Standards von ARD, ORF, SRF und ZDF*. Retrieved June 29, 2017, from https://www.zdf.de/barrierefreiheit-im-zdf/untertitel-standards-von-ard-orf-srf-und-zdf-100.html

# Appendix A: Abstract in English and German

Over the past few years, subtitles have gained greater visibility with the public due to increasing digitalisation and internet use. Consequently, they are no longer confined to traditional spheres like films and TV and are increasingly a part of people's everyday experience. Yet, their variable quality and inconsistencies between translations of the same source material, not only in films and series, but also in e-learning, scientific, and marketing videos, demand for an investigation of the problem and its resolving.

The above observation is the starting point for the hypothesis presented in this thesis, namely, that there is a need for computer-aided translation (CAT) tools in subtitling environments. This thesis, therefore, designs a solution aimed at tackling translation problems in both intralingual and interlingual subtitling processes. The proposed solution aims to facilitate work for translators in the subtitling industry by simultaneously raising their productivity and output quality.

Section 1 presents essential information about current translation technology and audiovisual translation (AVT) as context for the hypothesis. That hypothesis is substantiated in Section 2 by arguments for such a tool from an academic and professional point of view, including some field tests of currently available tools. Section 3 identifies the software features required for my proposed solution and also examines useful solutions already on the market by analysing subtitle attributes and their translational needs. Section 4 experiments with the tools identified in Section 3 to suggest functionalities that a subtitling software should include in order to boost consistency, and thus quality, while at the same time increasing productivity for translators working in subtitling environments. My final conclusions are briefly presented in Section 5.

### *Deutsch*

In den letzten Jahren ist die Sichtbarkeit von Untertiteln in der Öffentlichkeit aufgrund von zunehmender Digitalisierung und Internetnutzung gestiegen. Daher sind sie vermehrt im Alltag anzutreffen, sei es auf Facebook oder Video-Plattformen. Dennoch signalisieren häufige Qualitätsabweichungen und inkonsistente Übersetzungen des gleichen Ausgangstexts in Filmen und Serien, aber auch in E-Learning-Kursen, wissenschaftlichen wie auch zu Marketingzwecken genutzten Videos einen Handlungsbedarf dafür, die Gründe zu erforschen und Lösungen zu ermitteln.

Diese Beobachtung stellt den Ausgangspunkt für die Hypothese in dieser Masterarbeit dar, dass der Einsatz von computergestützter Übersetzungstechnologie, sogenannten CAT-Tools, im Bereich der Untertitelung notwendig ist. Die vorliegende Arbeit beschäftigt sich daher mit dem Entwurf einer technischen Lösung für das Problem der inkonsistenten Übersetzung von Untertiteln. Diese Lösung hat zum Ziel, Arbeitsvorgänge für intra- wie auch interlinguale Untertitler_innen zu erleichtern und gleichzeitig ihre Produktivität wie auch Produktqualität zu steigern.

Abschnitt 1 liefert Grundlageninformationen über derzeitig erhältliche Übersetzungstechnologien und audiovisuelles Übersetzen. Die Hypothese wird in Abschnitt 2 untermauern, dass eine Notwendigkeit für eine zusammengeführte Lösung von Untertitelungs- und Übersetzungs-Software vorhanden ist. In Abschnitt 3 werden Komponenten identifiziert, die beim Übersetzen von Untertiteln notwendig sind, um nach erfolgter Analyse von Untertiteleigenschaften und deren impliziert notwendigen technischen Anforderungen zu untersuchen, ob eine Lösung dafür bereits auf dem Markt ist. Abschnitt 4 beschreibt und dokumentiert die durchgeführten Tests der gefundenen Programme und unterbreitet einen Vorschlag, welche Komponenten ein Untertitelungsprogramm mit Übersetzungselementen haben soll, um einheitliche Übersetzungen zu gewährleisten und so die Qualität und gleichzeitig die Produktivität der Untertitler_innen zu steigern. Die Schlussfolgerungen daraus werden in Abschnitt 5 zusammengefasst.

# Appendix B: Table 17 – Source text for field tests (1) and (3)

| **English original by L. Carroll** | **German translation by C. Enzensberger** |
|---|---|
| “After a pause, Alice began, ‘Well! They were BOTH very unpleasant characters –’ Here she checked herself in some alarm, at hearing something that sounded to her like the puffing of a large steam-engine in the wood near them, thought she feared it was more likely to be a wild beast. ‘Are there any lions or tigers about here?’ she asked timidly.<br>‘It's only the Red King snoring,’ said Tweedledee.<br>‘Come and look at him!’ the brothers cried, and they each took one of Alice's hands, and led her up to where the King was sleeping.<br>‘Isn't he a LOVELY sight?’ said Tweedledum.<br>Alice couldn't say honestly that he was. He had a tall red night-cap on, with a tassel, and he was lying crumpled up into a sort of untidy heap, and snoring loud …”<br>(Carroll, L. 1871: 79-80) | „Nach einigem Nachdenken sagte Alice: „Also, dann waren sie eben alle beide abscheuliche Gesellen –“; und damit hielt sie einigermaßen bestürzt inne, denn sie vernahm ein Geräusch, das ihr wie das Schnauben einer großen Lokomotive vorkam, ganz nah im Wald, und zugleich fürchtete sie, es könnte vielleicht von einem wilden Tier kommen. „Gibt es in dieser Gegend viele Löwen und Tiger?“, fragte sie zaghaft.<br>„Das ist nur der Schwarze König“, sagte Zwiddeldei. „Er schnarcht.“<br>„Komm und schau ihn dir an!“, riefen sie zu zweit, fassten Alice beiderseits an den Händen und führten sie zu dem schlafenden König.<br>„Sieht er nicht wunderhübsch aus?“, fragte Zwiddeldum.<br>Da hätte Alice freilich lügen müssen. Er hatte eine hohe schwarze Schlafmütze mit einer Quaste auf, lag zu einem unordentlichen Häuflein zusammengerollt da und schnarchte laut …“<br>(Enzensberger 1974) |

*Table 17: Lewis Carroll’s original of* Through the Looking Glass *vs. Christoph Enzensberger’s* Alice hinter den Spiegeln

# Appendix C: Table 18 – Auto-synchronisation results for field test (1)

| **Auto-sync'ed by YouTube** | **wpm** | **Manually time-coded in WinCAPS** | **wpm** |
|---|---|---|---|
| 0001 00:00:08:00 00:00:12:06 | 79 | 0001 00:00:08:00 00:00:10:28 | 84 |
| After a pause Alice began, "Well! | | After a pause Alice began, | |
| 0002 00:00:12:06 00:00:17:16 | 157 | 0002 00:00:10:29 00:00:15:05 | 130 |
| They were both very unpleasant characters | | "Well! They were both | |
| -" Here she checked herself in some alarm, | | very unpleasant characters -" | |
| 0003 00:00:17:16 00:00:22:22 | 171 | 0003 00:00:15:06 00:00:19:06 | 201 |
| at hearing something that sounded to her like | | Here she checked herself in some alarm, | |
| a puffing on a large steam-engine in the wood | | at hearing something that sounded to her | |
| 0004 00:00:22:22 00:00:29:07 | 101 | 0004 00:00:19:07 00:00:23:13 | 141 |
| near them, though she feared it was more likely | | like a puffing on a large steam-engine in the | |
| to be a wild beast. | | wood near them, | |
| 0005 00:00:29:07 00:00:34:07 | 124 | 0005 00:00:23:14 00:00:28:23 | 100 |
| "Are there any lions or tigers about here?" | | though she feared it was | |
| she asked timidly. | | more likely to be a wild beast. | |
| 0006 00:00:34:07 00:00:38:01 | 137 | 0006 00:00:28:24 00:00:34:14 | 111 |
| "It's only the Red King snoring," said Tweedledee. | | "Are there any lions or tigers about here?"<br>she asked timidly. | |
| 0007 00:00:38:01 00:00:43:15 | 155 | 0007 00:00:34:15 00:00:38:02 | 148 |
| "Come and look at him!" the brothers cried, | | "It's only the Red King snoring," | |
| and they each took one of Alice's hands, and | | said Tweedledee. | |
| 0008 00:00:43:15 00:00:47:22 | 95 | 0008 00:00:38:03 00:00:43:16 | 150 |
| led her up to where the King was sleeping. | | "Come and look at him!" the brothers cried, and they each took one of Alice's hands, | |

| Auto-sync'ed by YouTube | wpm | Manually time-coded in WinCAPS | wpm |
|---|---|---|---|
| 0009 00:00:47:22 00:00:52:17 | 92 | 0009 00:00:43:17 00:00:46:18 | 146 |
| "Isn't he a lovely sight?" said Tweedledum. | | and led her up to where<br>the King was sleeping. | |
| 0010 00:00:52:17 00:00:56:03 | 118 | 0010 00:00:49:25 00:00:55:17 | 150 |
| Alice couldn't say honestly that he was. | | "Isn't he a lovely sight?" said Tweedledum. Alice couldn't say honestly that he was. | |
| 0011 00:00:56:03 00:01:02:08 | 135 | 0011 00:00:55:18 00:01:04:10 | 81 |
| He had a tall red nightcap on, with a tassel, | | He had a tall red nightcap on, | |
| and he was lying crumpled up into a sort of | | with a tassel, and he was lying crumpled up | |
| 0012 00:01:02:08 00:01:09:04 | 49 | 0012 00:01:04:11 00:01:07:24 | 132 |
| untidy heap, and snoring loud ... | | into a sort of untidy heap, and snoring loud ... | |

*Table 18: Evaluation of Auto-synchronization using wpm as a factor*