Movie Subtitles at the Service of Natural Language Processing

Jéssica Cristina Azevedo Veiga
jessica.veiga@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

August 2020

Abstract

The appearance of the *world wide web* allowed the creation and distribution of various extensive corpora. One such corpora are movie and TV show subtitles used in natural language processing to perform tasks such as statistical analysis, conversational agents, among others. However, majority of researchers first need to subject the subtitles to their own preprocessing steps to create a corpus suitable to their task requirements. B-Subtle is an open source framework including various preprocessing steps to build personalized dialogue corpora. In this work we have extended that framework to incorporate additional preprocessing steps considering the existence of subtitle duplicates, the frequency of dialogue turns and the presence of conversation topics. Our goal is to potentially help researchers avoid having to repeatedly implement preprocessing steps and to reduce their corpus to a more manageable size, requiring less computational power and storage capacity, while still being tailored to their system requirements. Furthermore, in this work we also developed from scratch B-Subtle App, a cross-platform desktop application executing the B-Subtle framework in the background and offering statistical analysis of the produced corpora in the form of a visual dashboard using state-of-the-art techniques in the field of information visualization. Our subsequent goals are to offer researchers a visualization that can potentially help in the incremental process of discovering the optimal sequence of preprocessing steps according to their task requirements through the iterative evaluation of the produced corpora and to attempt extending the usage of B-Subtle to a broader audience additionally including cinematographic enthusiasts.

Keywords: Subtitles, Preprocessing, Dialogue Corpora, Statistical Analysis, Visual Dashboard

1. Introduction

The *world wide web* makes it possible for people all around the world to directly and indirectly contribute to the extension of an ever growing collection of publicly available data. Consequently, there is an enormous amount of research works taking advantage of such unlimited data sources, in particular movie scripts and subtitles to perform various tasks from providing more resources to the scientific community [20], to statistical analysis [2], [21], machine translation [14], creation of knowledge-bases [25], movie summarization [13], violence prediction [16], conversational agents [3], [11], [27], [1], among others.

In all those research works, movie scripts and subtitles are subject to preprocessing phases to build personalized dialogue corpora specific to accomplish the intended task. That preprocessing phase includes multiple steps such as tokenization, normalization, semantic similarity filtering, movie genre filtering, advertisements removal, named entity replacement, just to name a few. Because movie scripts and subtitles constitute a case of overly large corpora, the implementation of preprocessing steps becomes imperative to process it in computers with less computational power and storage capacity. Moreover, because every time researchers implement their own preprocessing phase, in most cases they implement similar or exact same preprocessing steps, this alerted us to the fact that it would be helpful to merge a variety of useful and commonly implemented preprocessing steps in a single tool made available to the scientific community.

B-Subtle[1] is an open source framework that allows to automatically build personalized dialogue corpora containing dialogue turns extracted from subtitles belonging to the OpenSubtitles Corpus [14]. The framework relies on the existence of a pipeline that merges various preprocessing steps and contains an analytics module to provide infor-

[1] https://gitlab.hlt.inesc-id.pt/jcav/b-subtle_v2.0.

mation on the pipeline performance.

In this work we propose to extend the original version of B-Subtle to include some of the preprocessing steps being used in the scientific community, such as subber annotation removal, as well as others required by the HLT community[2], such as topic filtering, as means to support a broader set of research tasks and develop from scratch a cross-platform desktop application presenting a visual dashboard for statistical analysis not only on the pipeline performance but also the produced dialogue corpora as an alternative to the original sole console-based interface.

This work list of contributions include:

- **Extension of B-Subtle framework** with the improvement of existent and creation of new pipeline components as well as analytics modules.
- **Creation from scratch of a statistical analysis visualization** as a desktop cross-platform application for the subsequent exploration of produced corpora.
- **Journal Paper** "The B-Subtle framework: tailoring subtitles to your needs", Miguel Ventura, Jéssica Veiga, Luísa Coheur and Sandra Gama, Language Resources and Evaluation (in press).

All code implemented during the course of this work is publicly available on GitLab.

2. Related Work

There are various research works producing dialogue corpora based on movie scripts, such as Movie-DiC Corpus [2], Cornell Movie-Dialogue Corpus [7], MovieTriples Corpus [24] and Filtered Movie Script Corpus [20], as well as, based on movie subtitles, such as SubTle Corpus [1] and OpenSubtitles Corpus [14], among others [23]. Those corpus were subject to various preprocessing steps including metadata enrichment, tokenization, named entity recognition, normalization, semantic similarity filtering, advertisements removal, among others.

There are also research works such as IRIS [3], Joker Chatterbot [11], TickTock [27], that on the other hand develop CAs that use a similarity metric to find the closest match to a user utterance, known as *trigger*, in a knowledge-base composed of further preprocessed versions of such dialogue corpora, in order to return as response the corresponding *answer* and form an *interaction pair*. Examples of such further preprocessing steps include misspelled words removal, replacement of named entities with custom tags, movie genre filtering, subtitle creator annotations removal, stopwords removal, among others.

Moreover, even though after thorough investigation we were not able to find any framework targeted to preprocess movie subtitles as B-Subtle, we still found a research that merges a variety of preprocessing steps specific to Twitter corpora in a single tool made available to the scientific community called TWORPUS [4]. This tool allows researchers to filter tweets based on various metadata such as IDs of both the tweet and user, word and character counts, hashtags, among others.

All of the above researches prove that usually movie scripts and subtitles are not used in their raw form, but rather subject to a diversity of preprocessing phases that vary according to the researchers specific tasks. And, for that reason they are of particular interest to us, since they point out the need to extend B-Subtle framework with further preprocessing steps to support an even broader set of research tasks.

Furthermore, besides the production and usage of dialogue corpora for various research purposes another important task is the analysis of the produced corpora. Even though after thorough investigation we were not able to find any tool that analyses movie subtitles, we still found various research works that allow to analyse news, tweets, blog posts, books corpora through a visual interactive dashboard focusing on three major text analytics tasks such as word frequency (in FinanViz[18], Compare Clouds[8] and Mitchell WordCloud[22]), sentiments (in FinaVistory [6] and PEARL[28]) and topics (in Parallel Topics[9] and LeadLine[10]). Some characteristics of those tools consist in following a *drill-down approach*, offering the possibility to transition between overview and details of raw text, using *multiple coordinated views*, with the use of various techniques including brushing, highlighting, selecting, searching and filtering, in the case of tools focused on word frequency analysis the tendency to use *tag clouds* and on topic analysis the tendency to use *streamgraphs*. Besides those characteristics, some disadvantages include occlusion, due to large datasets, violation of Weber's Law [12], due to placement of data to be compared farther apart from each other, as well as, violation of other principles such as Expressiveness[15], Tufte "Labeling should be clear and detailed"[26] and Nielsen "Recognition rather than recall" and "Provide help and documentation"[19].

Therefore, all of the above works prove that it is also indispensable to use state-of-the-art visualization techniques to help researchers improve their efficacy and efficiency in the analysis of the produced corpora. And, for that reason, they are

[2] https://www.hlt.inesc-id.pt/

of particular interest to us, since they point out the need to improve B-Subtle interface and present us an insightful guide to sketch our visual analytic tool.

3. Methodology

In this work we extended the original version of B-Subtle framework with additional pipeline components inspired from some preprocessing steps of the works described in previous section and others required by the HLT[3] community, which will be described in subsection 3.1, as well as, improved its sole console-based interface with an alternative cross-platform desktop application developed from scratch, which will be described in subsection 3.2.

3.1. Pipeline Components

This work update to the **B-Subtle Framework** include the addition of entire new pipeline components to serve new preprocessing steps for the creation of an even more personalized subtitle corpora. Some of those new preprocessing steps are responsible for reducing the volume of the starter subtitle collection to improve time efficiency of the pipeline, others are responsible for gathering more information on the subtitle corpora, while the remaining are responsible for increasing the number of filtering options to choose from when outlining the process for the creation of custom subtitle corpora. Those new pipeline components are described next in greater detail.

3.1.1 Subtitle Duplicates Removal

B-Subtle supports preprocessing subtitles from OpenSubtitles Corpus, which for each natural language and each movie or TV show, in most cases, includes more than one subtitle leading to the existence of *subtitle duplicates*, which can be misleading for some research tasks and have bad impact on time efficiency of the preprocessing pipeline. Therefore, we implemented the pipeline component **Subtitle Duplicates Cleaner**, which is responsible for removing duplicated subtitles from the starter OpenSubtitles Corpus using metadata information to filter the subtitle with either the highest user rating - using tag $<$*rating* $>$ in their XML format - or the highest machine rating using a custom rating algorithm based on the linear combination of subtitle completeness and accuracy heuristics - using tags $<$*sentences* $>$, $<$*corrected_words* $>$ and $<$*unknown_words* $>$ in their XML format. In the later, the rationale is that for a given subtitle the higher the number of utterances and corrected words and the lower the number of unknown words, the higher will be its rating.

3.1.2 Frequent Pairs Extraction

Subtitle corpora composing dialogues preserve an enormous diversity of situational contexts, which differ in their frequency of occurrence from common daily life greetings to rare dialogues such as *"Ogres are like onions."*(*Shrek, 2001*). This distinction could be useful in particular research cases such as training a CA, which would benefit from the use of a knowledge-base with frequent interaction pairs rather than rare ones. Therefore, we implemented the pipeline component **Frequency Interaction Filter**, which is responsible for filtering interaction pairs based on the frequency of occurrence of their trigger, answer or both, in the starter corpora per subtitle. In obtaining the count for that frequency of occurrence we had to consider the fact that since interaction pairs are overlapping pairs of subsequent utterances, the frequency should be reduced in half for all utterances except the very first and last of a subtitle.

3.1.3 Movie Topic Filtering

Another consequence of the enormous diversity of situational contexts present in subtitles is the insane multitude of underlying conversation topics, which for target domain research tasks can constitute a problem. Therefore, we implemented the pipeline component **Topics Metadata Filter**, which is responsible for filtering subtitles based on their underlying conversation topics, using the corresponding movie/TV show plot-keywords available online in the TMDb official website[4]. To achieve this filtering we had to account for the occurrence of compound nouns, as well as, inflected and derived word forms, which resulted in the usage of a word stem inclusion check between the subtitle list of plot-keywords and the user provided list of required topics. In addition, we also accounted for the user intention to search for multiple topics with possibility to specify if the subtitle list of plot-keywords should include at least one or all the user required topics.

3.1.4 Pair Conversation Topic Filtering

Because the previous component results in a coarse filtering of topics based on subtitles, meaning possibly results in a set of dialogues that despite including the required conversation topics still have a high probability of including many others, it was required to provide a complementary refined filtering of topics based on interaction pairs. Therefore, we implemented the pipeline component **Topics Interaction Filter**, which is responsible for filtering interaction pairs based on their underlying

[3] `https://www.hlt.inesc-id.pt/`

[4] https://www.themoviedb.org/

conversation topics inferred through the vocabulary present in their composing utterances. However, the user list of topics is limited and does not provide full coverage of the diverse and extensive vocabulary of the targeted conversation topic, which could lead to missing dialogues addressing the same conversation topic, but using different vocabulary terms. Therefore, we also implemented the pipeline component **Conversation Filter**, which is responsible for filtering sequences of a user-defined number of subsequent interaction pairs neighbouring the ones previously filtered by the Topics Interaction Filter or any filter under the category of Interaction Filters.

3.2. Visual Analytic Tool

This work improvement to the B-Subtle sole console-base interface includes the development from scratch of an entire visual analytic tool consisting on a cross-platform desktop application called **B-Subtle App**. Since this application presents in the form of a visual dashboard statistical data concerning a variety of metrics related not only with the execution of the preprocessing pipeline, but also with the produced subtitle corpora, it can both help researchers during their incremental process of discovering the optimal preprocessing pipeline through the iterative evaluation of the produced dialogue corpora, as well as, expand the usage of the wrapping B-Subtle framework to a broader audience not only researchers and not strictly familiar with operating a system console.

To ensure the B-Subtle App fits the requirements of its targeted audience, during its development we followed a *top-down* approach, in which we first focused on gathering a clear definition of the tasks and questions its visual dashboard needed to provide answer, followed by the implementation of the final functional prototype with idioms that attempt to provide answer to those exact questions.

3.2.1 Task Abstraction

In the following list we present the set of tasks and corresponding example questions our tool proposes to answer.

- **Task 1 – Summarize pipeline performance metrics.** What is the total pipeline execution time?
- **Task 2 – Compare pipeline performance metrics.** Have metadata filters discarded more subtitles than interaction pair filters?
- **Task 3 – Summarize words/topics occurrence in screenplays.** Is the topic “alien invasion” more frequent than “terrorism” in screenplays?
- **Task 4 – Compare words/topics occurrence in screenplays between groups.** How different are the vocabularies used in horror and comedy movies? And in the 90’s and recent movies?
- **Task 5 – Identify occurrence of special screenplay words.** What are popular character names used in screenplays?
- **Task 6 – Summarize sentiments occurrence in screenplays.** In general, are movies more positive or negative?
- **Task 7 – Compare sentiments occurrence in screenplays between groups.** Are crime movies more negative than horror movies?

3.2.2 Functional Prototype

The B-Subtle App was developed as a desktop application available for Windows, Mac and Linux operating systems, built with Electron, Vue, Bootstrap and Highcharts web technologies and executing in the background the most recent version of the B-Subtle framework.

The B-Subtle App is initiated with a *landing page*, where the user is prompted to choose between preprocessing a user-defined sample of the OpenSubtitles Corpus to create new personalized corpora with further analysis, corresponding to the **Configuration File** option, or, alternatively, simply analyse personalized corpora created beforehand, corresponding to the **Analytics Folder** option. If the user selects the former, it will be required to upload a starter configuration file and afterwards the application will navigate to a *loading page* which displays the progress in the execution of the preprocessing pipeline by the framework, displaying a success or failure message with the possibility to open the file system explorer on the folder containing the recently generated corpora once the execution is over. If the user selects the later, it will be required to upload the analytics folder generated by the framework during preprocessing and creation of a given personalized corpora beforehand and, afterwards, the application will immediately navigate to the analytics dashboard presented in Figure 1.

The application analytics dashboard consists in a **customizable dashboard**, composed by a left side **control panel** and a right side collection of available **widgets**.

The control panel is composed of a **search section**, which includes a *search box* that at the present moment only allows to search for available widgets, a **widget section**, which allows to toggle on and off widgets, a **configuration section**, which allows to choose different data ag-

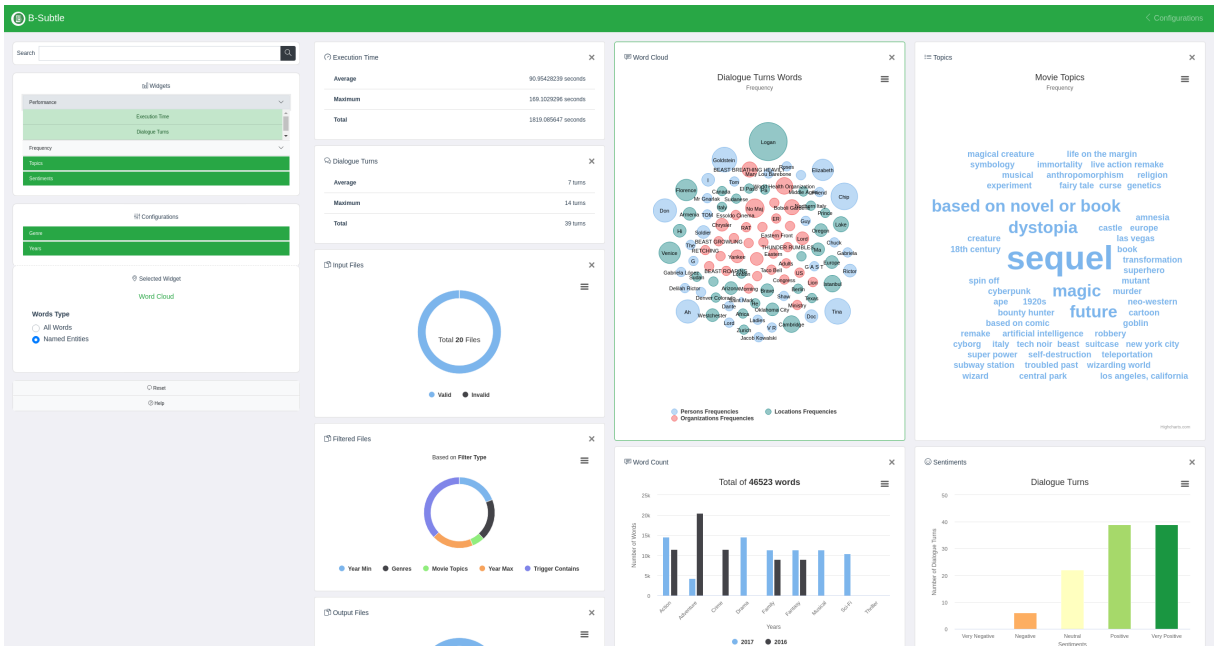

Figure 1: B-Subtle App analytics dashboard.

gregation scopes and methods with **global** and **specific configurations**, and a **support section**, which includes a **reset** button, to return the application to its initial state including the creation of all the available widgets that were closed during the user interaction and the adjustment of the global and specific configurations of each widget to its default values, and a **help** button, to present an application help guide documentation popup with a detailed guide including an introduction on the development context of the application, information on how to download a sample of the OpenSubtitles Corpus and create a starter configuration file, information on all the different functionalities available in each of the control panel sections, and, finally, information on extra functionalities available in each widget of the dashboard.

In what concerns the widget collection, widgets are grouped according to the subject matter of their metrics, meaning into four major views: **performance** providing answer to Task 1 and 2; **words** providing answer to Tasks 3, 4 and 5; **topics** providing answer to Tasks 3 and 4; and **sentiments** providing answer to Task 6 and 7. In more detail, the available widgets are the following:

- **Execution Time** – falls under the performance view and encodes both the total execution time, as well as, the average and maximum execution time per subtitle file in the starter subtitle corpora, using as unit seconds.
- **Dialogue Turns** – falls under the performance view and presents both the total number of filtered dialogue turns, as well as, the average and maximum number of filtered dialogue turns per each file of the starter subtitle corpora.
- **Input Files** – falls under the performance view and encodes the volume of the files composing the starter subtitle corpora emphasizing the validity and invalidity of such files according to their conformance with the OpenSubtitles Corpus XML standard schema, using as unit either number of files or Megabytes.
- **Filtered Files** – falls under the performance view and encodes the number of files accepted per each type and instance of filter composing the preprocessing pipeline defined by the user.
- **Output Files** – falls under the performance view and encodes the volume of the files composing the final produced corpora differentiating between the four different produced file formats supported by the framework including JSON, Legacy, Parallel and XML, using as unit either the number of files or Megabytes.
- **Word Count** – falls under the words view and encodes the frequency of occurrence of 100 words present in the subtitle dialogue turns composing the final produced corpora with possibility to filter solely named entities and aggregate them in Persons, Locations and Organizations.
- **Word Cloud** – falls under the words view and encoding the number of words present in the subtitle dialogue turns composing the final

produced corpora per movie genre, per production year or both simultaneously.

- **Topics** – falls under the topics view and encodes the frequency of occurrence of the plot-keywords from the movies present in the final produced corpora, extracted from the TMDb.

- **Sentiments** – falls under the sentiments view and encodes the total number of dialogue turns present in the final produced corpora classified either as very negative, negative, neutral, positive or very positive, per movie genre, per production year or in total.

Moreover, still concerning the widget collection however changing the perspective to its interactivity, besides the **re-encoding** and **re-configuration** possible through the usage of the global and specific configurations available in the control panel, in each widget there are extra functionalities that were implemented and allow the user to further interact with the corresponding idioms, such as *mouse hover* over a given data instance in the idiom, which will render a **tooltip** with detailed information on the attribute-value pairs of the tracked data instance, *mouse click* on a given **legend** in the idiom, which results in filtering out all the data instances falling under such attribute-value pair group corresponding to the tracked legend, and, finally, the possibility to resize the idiom to **fullscreen mode**, with the purpose of avoiding any possible data occlusion due to screen size limitations, as well as, the possibility to **print** and **download** idioms either in *PNG*, *JPEG*, *PDF* or *SVG* file formats, with the purpose of helping the user persist its explorations and discoveries made with the analytics dashboard for reporting reasons.

4. Evaluation

This work evaluation encompassed both *usability* and *utility testing*. For each of those tests, we defined a different set of questions to be answered by the users with the help of our tool. Those sets of questions attempted to cover all the tasks described in subsubsection 3.2.1 and included simple questions involving a single task and more complex questions involving more than one task. We have conducted those tests with both NLP researchers from the HLT community[5], students from the IST university[6] and personal acquaintances, all with high interest in cinematography, as well as, familiarized and comfortable with the use of technologies. Furthermore, we mainly deployed observation and survey data collection methods for the tests with our users.

[5] https://www.hlt.inesc-id.pt/

[6] https://tecnico.ulisboa.pt

4.1. Usability

Performing usability tests enabled us to ascertain how effectively and efficiently the users can accomplish tasks with the help of our tool and their overall satisfaction. Even though our objective was to follow an *iterative design process*, we were only able to perform a single iteration consisting in the *summative evaluation*. We performed this evaluation with 15 users through social communication platforms over mainly voice calls and, in some cases, also video chats. We have adopted a *quantitative testing* approach by measuring the time required to answer each question, as well as, the number of actions that did not contribute to answer the question. In Table 1 we present the means and standard deviations for both time and errors measured during the users usability evaluations. From those statistics we can notice that question Q2 proved to have the highest complexity given its higher mean for both time and errors in comparison with the remaining questions.

ID	Time		Errors	
	Mean	Std. Deviation	Mean	Std. Deviation
Q1	48.13	53.116	0.153	0.594
Q2	133.20	43.080	0.215	0.834
Q3	67.13	38.706	0.165	0.640
Q4	27.67	15.783	0.159	0.617

Table 1: App evaluation questions definition and descriptive statistics.

Provided that as mentioned before each question had a different level of complexity, it is also interesting to further understand if the different question complexities impacted the variables time and errors. To derive conclusions on that impact we conducted various statistical tests starting with the **Shapiro-Wilk** test, from which we concluded both time and errors were not normally distributed, from results shown in Table 2, followed by the application of **Kruskal-Wallis** test, from which we concluded there was a difference in the means of both time and errors depending on the question complexity, from results shown in Table 3.

Variable	Question	Shapiro-Wilk Sig.
Time	Q1	0.000
	Q2	0.061
	Q3	0.458
	Q4	0.009
Errors	Q1	0.000
	Q2	0.015
	Q3	0.000
	Q4	0.000

Table 2: App evaluation Shapiro-Wilk results for $\alpha = 0.05$.

Variable	Kruskal-Wallis H
Time	28.843
Errors	21.411

Table 3: App evaluation Kruskal-Wallis results for critical value $Q = 7.81473$.

Still in the usability evaluation, we asked users

to answer the classic 10 questions of the SUS, we used the method defined in [5] to measure the usability of our tool and achieved a final score of 85.83, which given its greater then 80.3 means our tool achieved an excellent usability.

4.2. Utility

Complementary to usability, performing utility tests will enable us to ascertain if our tool offers all the features required by our target users and possibly more [17]. These tests were performed at the end of the tool development phase and involved a smaller number of 2 users per each group of researchers and cinematographic enthusiasts. These users had also participated in the previous usability evaluation, provided it was required that they had some *context knowledge*. During those tests we considered subjective metrics by adopting a **thinking aloud testing** approach with no time restrictions and live qualitative comments. As a result we gathered a rich collection of not only user constructive critiques to implemented features, but also our examiner observations on usage patterns, which will both be described next.

In the context of **researchers**, one of the most predominant critique was the need to provide high-level description on the information being visualized in each widget. In addition we also gathered an interesting observation on usage patterns which occurred with both users and consisted in them removing all the widgets from the dashboard that did not seem related with the problem being solved prior to any further problem-solving step.

In the context of **cinematographic enthusiasts**, users predominant critiques concerned the design inconsistency between the global and specific configurations which might lead the user to misleadingly perceive them as different forms of control, the difficulty in noticing the border highlighted in selected widgets, the hidden position of the reset button on some screen sizes, as well as, the difficulty in comparing word frequencies solely through bubble radius in the Word Cloud widget. Additionally, we also gathered observations on usage patterns and found that most users overlooked the existence of interactive legends for some widgets that could be toggled on/off to reconfigure the visualization.

Some observations commonly found on **both user targeted groups** were the frequent misunderstanding between the widget title and subtitle, as well as, the highly frequent case in the beginning of the test when users used the widget section available in the control panel to toggle on/off widgets from the dashboard, but due to screen size limitations and scrolling not being an intuitive action for most users it was not perceptible to them that widgets were being added and removed. The latter leads to another highly frequent critique regarding the additional difficulty in understanding when a given widget was selected and that upon selection there existed a cause-effect relationship between the selected widget and the content of the specific configuration section available in the control panel.

4.3. Discussion

This work evaluation proved the difference in usage of our application for researchers and cinematographic enthusiasts, since each user group showed slightly different concerns during their interaction with our tool. Researchers mainly focused on features that could provide a precise description on the information being displayed on each widget, reflecting their concern not only on the value the application could provide in performing their research tasks, but also the need to have a deeper understanding of the application with easy to access memos in order to speed the learning curve and quickly formulate a mental model on the usage of the tool. In opposition, cinematographic enthusiasts mainly focused their attention on visual aspects of the visualization with frequent comments on design choices.

Moreover, our usability evaluation results determined our tool achieved a good usability score and there was a strong correlation between the task complexity and the time required to perform such task, as well as, the possible number of errors committed when performing it. However, in an attempt to further correlate the results of both usability and utility evaluations we can justify such correlation due to few inconsistencies and information gaps found in our tool by users throughout the tests. For instance multiple users claimed to had difficulty in finding the location of the widget specific configuration mainly due to its placement farther away from the corresponding widget, proving to be the cause for the increase in mean time and error rate for question Q2 in Table 1. In addition, we can also argue that such correlation might have been affected by the order in which questions were performed. For instance questions Q3 and Q4 both required the use of global configurations, however question Q3 marked the first user interaction with such feature and was immediately followed by question Q4 where users easily recalled that feature placement and behaviour from shot-term memory, contributing to the lowest mean of time when compared with the remaining questions.

5. Conclusions

B-Subtle is an open source framework developed in the context of a previous dissertation work, that allows researchers to automatically personalize corpora containing dialogue turns extracted from

subtitles belonging to the OpenSubtitles Corpus. However, its original version needed further extension of its preprocessing steps catalogue and implementation of an alternative to its sole console-based interface. Those combined determined the following fundamental goals and solutions of our present work.

Our ultimate goal is a framework to attempt to assist researchers working with subtitles corpora in the creation of personalized dialogue corpora specific to fit their research requirements, as well as, to assist them to avoid having to repeatedly implement the same preprocessing steps and to reduce the corpora to a more manageable size. For that reason, we have extended the B-Subtle framework with additional preprocessing steps which offer the possibility to reduce the volume of the starter subtitle corpora through the removal of subtitle duplicates, to filter dialogue turns based on the frequency of occurrence of their composing trigger, answer or both, to filter both subtitles and dialogue turns based on the presence of specific topics, as well as, to further extract conversations from the sequence of dialogue turns.

The secondary goal of our work is not only to further attempt to assist researchers in the incremental process of discovering the optimal sequence of preprocessing steps for their research through visualizations on information of the produced dialogue corpora, but also extend the usage of B-Subtle to a broader audience of cinematographic enthusiast apart from researchers. For that reason, we have developed from scratch B-Subtle App, a cross-platform desktop application powered by B-Subtle Framework and which is responsible for rendering a visual dashboard that aims to provide answer to various questions including metrics on the framework performance, as well as, on words, topics and sentiments frequency of occurrence, among others.

To determine if B-Subtle App offers in an effective and efficient manner all the features our target users require for the accomplishment of their tasks, we conducted a usability and utility evaluations which allowed us to conclude that our application achieved a good usability score of 85.83, as well as, the existence of a correlation between tasks complexity and both time spent and errors committed in performing such task. Moreover, allowed us to gather user constructive critiques of implemented features and usage patterns. These offered us insights on the concerns of each user group, with researchers mainly prioritizing effectiveness and efficiency while cinematographic enthusiasts mainly prioritizing user experience, as well as, on possible causes for the unexpected higher/lower complexity.

References

[1] D. Ameixa and L. Coheur. From subtitles to human interactions : introducing the SubTle Corpus From Subtitles to Interactions- Response pairs. pages 1–4, 2013.

[2] R. E. Banchs. Movie-DiC: a Movie Dialogue Corpus for Research and Development. *Jeju, Republic of Korea*, (July):203–207, 2012.

[3] R. E. Banchs and H. Li. IRIS: a Chat-oriented Dialogue System based on the Vector Space Model. *Proceedings of the 50th Annual Meeting ofthe Association for Computational Linguistics*, (July):37–42, 2012.

[4] A. Bazo, M. Burghardt, and C. Wolff. TWORPUS - An easy-to-use tool for the creation of tailored twitter corpora. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8105 LNAI:23–34, 2013.

[5] J. Brooke. SUS: A quick and dirty usability scale. *Usability evaluation in industry*, 189, 1996.

[6] Y. Y. Chan and H. Qu. FinaVistory: Using Narrative Visualization to explain social and Economic relationships in financial news. *2016 International Conference on Big Data and Smart Computing, BigComp 2016*, pages 32–39, 2016.

[7] C. Danescu-Niculescu-Mizil and L. Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. *CoRR*, abs/1106.3, 2011.

[8] N. Diakopoulos, D. Elgesem, A. Salway, A. Zhang, and K. Hofland. Compare Clouds : Visualizing Text Corpora to Compare Media Frames. *Proceedings of IUI Workshop on Visual Text Analytics*, pages 193–202, 2015.

[9] W. Dou, X. Wang, R. Chang, and W. Ribarsky. ParallelTopics: A probabilistic approach to exploring document collections. *VAST 2011 - IEEE Conference on Visual Analytics Science and Technology 2011, Proceedings*, pages 231–240, 2011.

[10] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou. LeadLine: Interactive visual analysis of text data through event identification and exploration. *IEEE Conference on Visual Analytics Science and Technology 2012, VAST 2012 - Proceedings*, pages 93–102, 2012.

[11] G. D. Duplessis, V. Letard, A.-L. Ligozat, and S. Ross. JOKER CHAT- TERBOT: RE-WOCHAT 2016 - SHARED TASK CHATBOT DESCRIPTION REPORT. *Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents - Development and Evaluation*, 2016.

[12] G. Fechner. Elements of psychophysics. Vol. I. 1966.

[13] P. J. Gorinski and M. Lapata. What's This Movie About? A Joint Neural Network Architecture for Movie Content Analysis. pages 1770–1781, 2018.

[14] P. Lison, J. Tiedemann, and M. Kouylekov. OpenSubtitles2018: Statistical Rescoring of Sentence Alignments in Large, Noisy Parallel Corpora. pages 1742–1748, 2018.

[15] J. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.*, 5(April 1986):110–141, 1986.

[16] V. R. Martinez, K. Somandepalli, K. Singla, A. Ramakrishna, Y. T. Uhls, and S. Narayanan. Violence Rating Prediction from Movie Scripts. 2019.

[17] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[18] N. V. Nguyen, V. T. Nguyen, V. Pham, and T. Dang. FinanViz: Visualizing Emerging Topics in Financial News. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, pages 4698–4704, 2019.

[19] J. Nielsen. Enhancing the explanatory power of usability heuristics. *Conference companion on Human factors in computing systems - CHI '94*, page 210, 1994.

[20] L. Nio, S. Sakti, G. Neubig, T. Toda, and S. Nakamura. Conversation dialog corpora from television and movie scripts. In *2014 17th Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA)*, pages 1–4, 2014.

[21] G. H. Paetzold and L. Specia. Collecting and Exploring Everyday Language for Predicting Psycholinguistic Properties of Words. *Proceedings of the 26th International Conference on Computational Linguistics*, pages 1669–1679, 2016.

[22] M. M. Schwarz, K. Marrio, and J. Mccormack. The Mitchell Library WordCloud: Beyond Boolean Search. In A. Bonnici, editor, *Proceedings of the 2017 ACM Symposium on Document Engineering*, pages 39–48. Association for Computing Machinery (ACM)", 2017.

[23] I. V. Serban, R. Lowe, P. Henderson, L. Charlin, and J. Pineau. A Survey of Available Corpora for Building Data-Driven Dialogue Systems. *CoRR*, abs/1512.0, 2015.

[24] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. *CoRR*, abs/1507.0, 2015.

[25] N. Tandon, G. D. Melo, and G. Weikum. Lights , Camera , Action : Knowledge Extraction from Movie Scripts. In *Proceedings of the 24th International Conference on World Wide Web*, pages 127–128, Florence, Italy, 2015. ACM.

[26] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1986.

[27] Z. Yu, Z. Xu, A. W. Black, and A. I. Rudnicky. TickTock RE-WOCHAT 2016 Shared Task Chatbot Description Report. *WHOCHAT workshop*, 2016.

[28] J. Zhao, L. Gou, F. Wang, and M. Zhou. PEARL: An interactive visual analytic tool for understanding personal emotion style derived from social media. *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014 - Proceedings*, pages 203–212, 2015.