

BECAM Tool - A Semi-automatic Tool for Bootstrapping Emotion Corpus Annotation and Management

*Slim Abdennadher¹, Mohamed Aly¹, Dirk Bühler²
Wolfgang Minker², Johannes Pittermann²*

¹Department of Computer Science, German University in Cairo, Egypt

²Institute of Information Technology, University of Ulm, Germany

slim.abdennadher@guc.edu.eg, mohamed.abdulazim@student.guc.edu.eg,

{dirk.buehler, wolfgang.minker, johannes.pittermann}@uni-ulm.de

Abstract

Corpus annotation is an important aspect in speech applications where stochastic models need to be trained and evaluated. Multimodal corpora are also annotated. Moreover, corpus annotation is an essential phase in the construction of emotion recognizer engines. Large corpora, as they are essential to construct representative knowledge bases, have been a problem for corpus annotators. Time consumed for labeling such corpora is very significant. Furthermore, manageability becomes more arduous and tedious. In this paper, we propose a semi-automatic tool, called BECAM tool, that will help corpus annotators in managing and annotating large sample emotion corpora.

Index Terms: Corpus annotation, emotion recognition, bootstrap

1. Introduction

The need for computers getting to “think” and “feel” is an issue which has been in research for many years. People are getting increasingly involved with computers daily, and they are looking for a more enriched interaction experience. When it comes to human beings, the fact that we understand each other and react to each other emotions makes our communication and interaction process very versatile. On the other hand, computers are still “emotionally challenged” [1]. Many studies have been conducted towards emotion recognition and its integration in computer applications. This integration might change the way a user interacts with the computer and provides the user with a new level of experience [2].

A variety of statistical approaches have been used for emotion recognition. Although there are other approaches, most of the work in this area [3, 4] rely on Hidden Markov Models (HMMs), which can be built and manipulated by the Hidden Markov Model ToolKit (HTK) [5]. For that, building a reliable model for emotion recognition requires a corpus with a large sample size. Sample sound files in a corpora need to be analyzed by the annotator and labeled accordingly to identify emotions within the sample. Then the labeled sample files are used as a knowledge base to train an emotion recognition engine. With an increasing sample size, the process of annotation becomes more time consuming and the manageability complexity increases significantly. Various approaches have been considered by corpus annotators to reduce the time needed for annotation. The bootstrap approach [6] is one of the used methods. The approach is intended to decrease the time needed by manual annotation by automatically generating annotations for unanno-

tated samples based on manually annotated samples. Automatically annotated samples are then manually corrected and added to the corpus, and the process proceeds iteratively until all samples are annotated. In this paper, we propose a semi-automatic tool based on HTK and a client-server architecture that enables corpus annotators to perform the bootstrapping on emotion corpora adding more flexibility, manageability, and reliability. We call the tool BECAM which stands for Bootstrapping Emotion Corpus Annotation and Management.

According to a survey conducted on existing annotation tools in [7], we came up to the conclusion that the main goal of the existing tools is to support various methodologies for corpus annotation. Analyzing the purpose of the tools surveyed, we found out that tools are designed for special or limited purposes. Various methods are used for annotation and various file and export formats are used among the tools. According to our knowledge, we consider our tool different from the existing ones since we do not physically deal with the annotation process, instead our focus is on automating the process by applying the bootstrap approach.

The paper is organized as follows. In Section 2 we introduce an example of the data used in emotion recognition as well as the annotation and we describe how the bootstrap process operates. In Sections 3 and 4 we describe the design, implementation and usage of the BECAM Tool, and in Sections 5 and 6 we evaluate the tool and discuss our future directions and intentions.

2. Processing Emotional Data

2.1. Example of the data and annotation

For our emotion recognition experiments the Database of German Emotional Speech of the Technical University of Berlin [8] has been used. This Database includes six of the basic emotions (German terms in brackets), namely anger (Wut), boredom (Langeweile), disgust (Ekel), fear (Angst), happiness (Freude) and sadness (Trauer) along with neutral recordings serving as references. Ten actors, 5 female and 5 male speakers, performed ten different everyday-speech utterances such as “The cloth is lying on the fridge” (“Der Lappen liegt auf dem Eisschrank”) or “She will hand it in on Wednesday” (“Das wird sie am Mittwoch abgeben”), each utterance in all emotions allowing a high comparability across emotions and speakers. As the recordings were made in an anechoic chamber, background noise could be minimized and therefore had no disturbing effect on the experiments. The emotional quality was rated by 20

persons, who were asked to listen to the utterances only once in front of a computer monitor and to assign the utterances to the different emotions as well as to specify how convincing the emotion was brought out [8]. Figure 1 shows the manual annotation using the HTK tool HSLab.

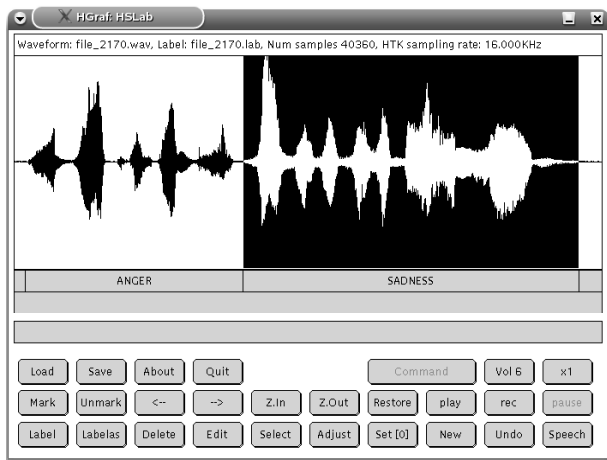


Figure 1: Manual annotation using HSLab

2.2. Bootstrap Process

The bootstrap process facilitates the manual labeling of samples in a corpus. At first, a small subset of samples is labeled manually. This labeled subset is then used to train a recognizer model, which is used to automatically recognize and annotate an unannotated subset in the corpus. The resulting subset from automatic annotation is then reviewed by the annotator for correctness and accuracy, and manually corrected if necessary. After modification and review, the samples of the automatically annotated subset are consistent with the previously labeled subset. A bigger subset containing the newly labeled and the previously used samples is now to be formed to construct a new recognizer model which is to be used to label another unlabeled subset of samples in the corpus. The process proceeds iteratively in the same way until all files are annotated in the corpus. Figure 2 illustrates the details of the bootstrap process [6].

3. Design and Implementation

3.1. Design

The tool provides the ability to define several corpora, import files, and export files based on the HTK formats. Several corpora can be useful for a team of annotators to work on several models simultaneously. It also provides features like playing

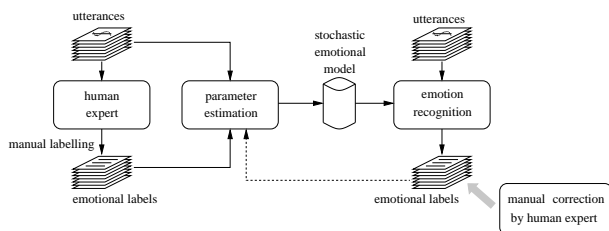


Figure 2: Illustration of the bootstrap process

files and displaying assigned annotations. Grammars and dictionaries used for an experiment can also be created and edited with the tool. Defined grammars and dictionaries can be saved and used for other experiments. Subsets of the corpus can be defined by selecting a group of files from the corpus. Different corpus' subsets can be used to define experiments with different training and testing sets. The tool provides the ability to experiment with the subsets. We define an experiment to be composed of a training set and a testing set which are non empty subsets of files from the corpus. An experiment can be performed to train a recognizer engine and test it on an existing subset which has been annotated. Results are then imported and saved. This can be helpful for annotators to decide for the reliability of a certain trained model. Experiments can also be performed to annotate unannotated samples. Samples in the training set are selected by the annotators analysis of the samples reliability according to his/her experience, or previous experiment results. Samples in the testing subset in this case are unannotated. The result file can then be exported to label files which will be reviewed and corrected by the annotators and finally imported to the database.

For usage simplicity, the tool provides a user friendly graphical user interface (GUI). Figure 3 shows a screenshot of the tool. The tool consists of a corpus, a grammar, a dictionary, a subset and an experiment manager. The corpus manager is used to import or export wave forms (WAV), label files (LAB) and feature (vector) files (MFV) to or from a corpus. It also provides functions like playing WAV files, displaying LAB files, and creating MFV files. It can also be used to define subsets by simple selections from the file list of the corpus. The grammar manager and dictionary manager are used to define grammars and dictionaries which are to be used in experiments on the corpus files. The experiment manager is used to define experiments, track current experiments by annotators, view results, and export experiment output files to label files. Predefined subsets, grammars and dictionaries act as bases for experiment setups. To enable multi-users, the tool is implemented based on a client-server architecture. This enables many annotators to work in parallel and hence speed up the process. This architecture is widely used by many annotation tools and was shown to work [9]. A centralized database acts as a repository holding corpora data, dictionaries, grammars, and experiment data. Experiments defined could be saved and executed later by the same or another annotator, or they could be immediately executed on the client side. The current status of an experiment is updated in the database. An experiment could be ready for execution, running, or completed (successfully or with an error). The name of the annotator and the machine name or IP address at which the experiment has been created or running on is also stored. Results of successfully completed experiments are updated in the database and can be viewed by various annotators. To avoid version conflicts, we currently rely on experiment and file import dates to the data base. Figure 5 shows a screenshot of the experiment manager.

3.2. Implementation

We have used Java for implementation which makes the tool operative in multi-environments since Java is a platform independent programming language. The database is implemented using the Java-based HSQL database (Hypersonic SQL, www.hsqldb.org). This database acts as the repository of corpus, dictionaries, grammars, subsets and experiment data. Having the data isolated from the application side makes it possible of having multiple client instances that accesses the same data

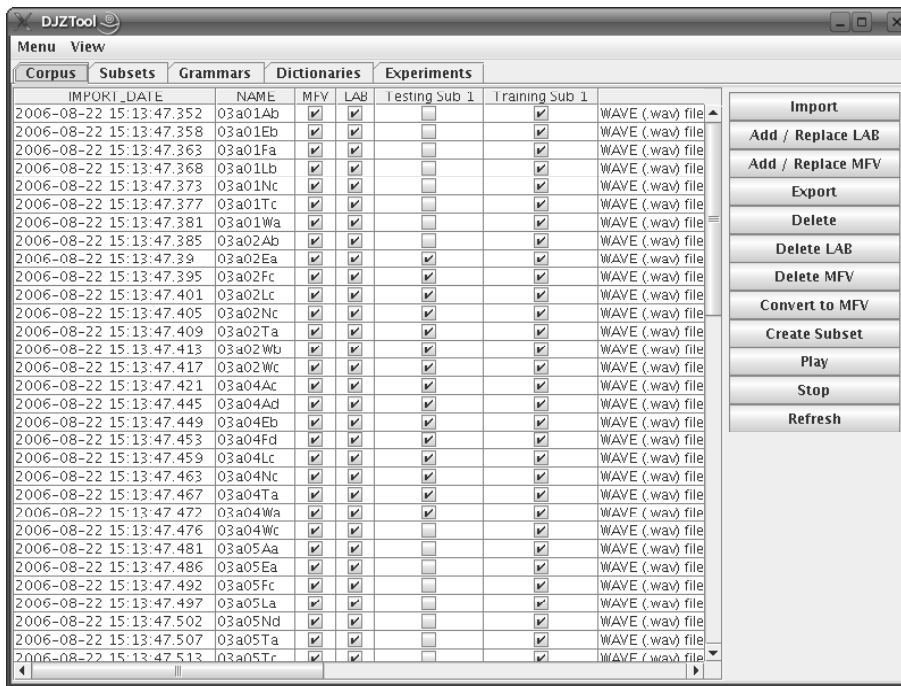


Figure 3: Screenshot of the BECAM Tool

at the same time and monitor experiment progresses. We use the HTK file formats as our main tool file import and export format. As described in [10], we have used the Praat tool [11] to extract emotion-relevant features from the waveforms. Among these features are pitch and formants, intensity, harmonicity, jitter and Mel-Frequency Cepstral Coefficients (MFCCs) plus their statistical computations. Figure 4 summarizes the tool structure.

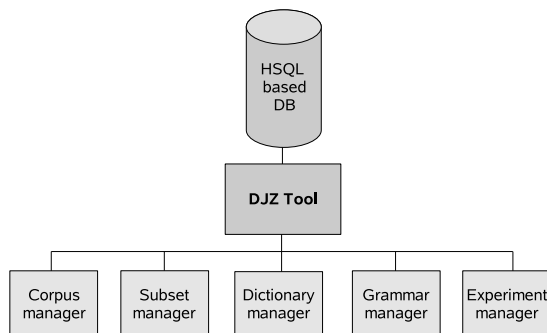


Figure 4: Structure of BECAM Tool

4. Using the BECAM Tool

Practically, the tool can be used to start with raw WAV files and help an annotator to end up with a representative corpus. In the beginning, an empty corpus is defined. WAV files are then imported to the corpus. The files are then converted to MFV files which will be used for feature extraction. A subset of the files is then annotated manually and the corresponding LAB files are imported. Experiments within the tool are then carried out by an annotator on the manually labeled subset until a good (de-

pending on experiment results or on the annotator's experience) representative subset is found. A labeling experiment using this subset is initialized to automatically annotate an unannotated subset of files within the corpus. The automatically generated annotations are then reviewed and corrected if necessary. Finally, the new annotations are imported and a new iteration can be started within the tool.

5. Evaluation of the BECAM Tool

We have tested the tool on the German Database of Emotional Speech [8] and on a small unlabeled corpus of speech data recorded from TV shows. With the aid of the training and testing interface and the subset selection tools the bootstrapping process has been simplified and accelerated significantly. Once a waveform is inserted in the database the respective feature file can either be added manually or the conversion can be performed from the GUI. For the automatic labeling, training and testing experiments are conducted on the selected training subset (files which are already manually labeled or corrected by the annotator) and testing subset (unlabeled files) using HTK. The recognizer results can then be corrected manually and added to the database. Especially the instantaneous indication which files are already labeled and which are missing allows the annotator to keep track of the progress of the labeling task. Due to the simplified structure of the tool the training period for novice annotators can be reduced by 60% still providing the annotator a very concise overview about the systematics of the bootstrap algorithm. Taking advantage of the multi-user capability of the tool and of the semi-automatic labeling, the amount of work can also be reduced depending on the previous knowledge of the annotator.

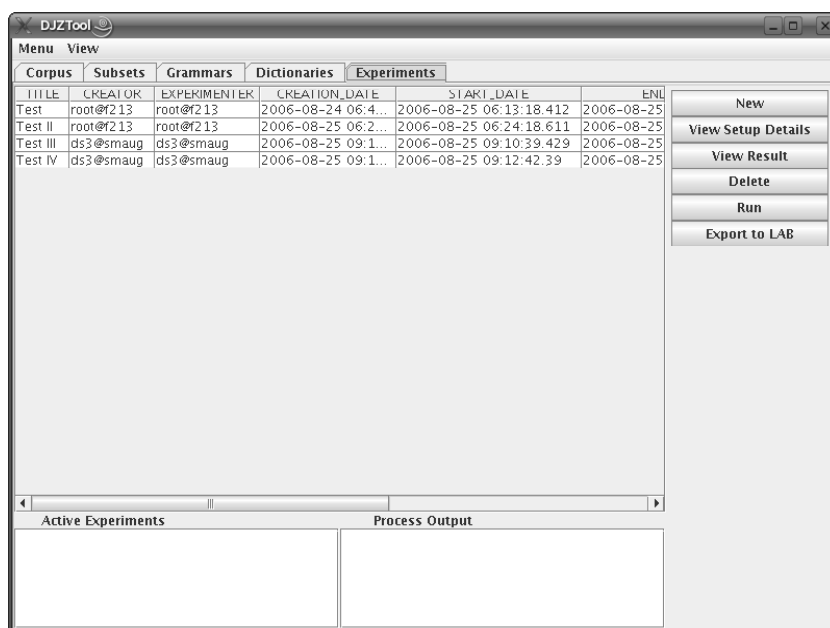


Figure 5: BECAM Tool Experiment Manager

6. Conclusion and Future Directions

In this paper, we have presented a semi-automatic tool that will help corpus annotators in managing and annotating large sample emotion corpora. Considering the advantages of the bootstrap process, time can be saved in labeling corpora with large sample sizes. The independency of the process on the corpora nature, makes the process applicable to other kinds of corpora. Currently, the BECAM Tool supports HTK formats and uses Praat tool for emotion-feature extraction. Various file formats could be considered in the future as well as various methods and tools for feature extraction. However, our tool can be integrated with existing tools which rely on the same formats. Manual annotation tools could also be integrated. We chose emotion corpora as a start, but we expect that we can support other types of corpora. Since we have only tested the tool on a small sized corpus, analysis of the feasibility and usability of the tool for large sized corpora is under investigation.

7. References

- [1] N. Sebe, M. S. Lew, I. Cohen, A. Garg, and T. S. Huang, "Emotion recognition using a Cauchy naive Bayes classifier," in *ICPR '02: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02) Volume 1*. Washington, DC, USA: IEEE Computer Society, 2002, p. 10017.
- [2] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human-computer interaction," *Signal Processing Magazine, IEEE*, vol. 18, pp. 32–80, 2001.
- [3] B. Schuller, G. Rigoll, and M. Lang, "Hidden Markov model-based speech emotion recognition," in *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 2 (ICME '03)*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 401–404.
- [4] L. C. DeSilva and P. C. Ng, "Bimodal emotion recognition," in *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*. Washington, DC, USA: IEEE Computer Society, 2000, p. 332.
- [5] S. Young, "The HTK Hidden Markov Model Toolkit: Design and philosophy," *Cambridge University Engineering Department, UK, Tech. Rep. CUED/F-INFENG/TR152*, 1994.
- [6] J. Pittermann and A. Pittermann, "Integrating emotion recognition into an adaptive spoken language dialogue system," in *2nd IET International Conference on Intelligent Environments*, Athens, Greece, July 2006.
- [7] L. Dybkær and N. O. Bernsen, "Towards general-purpose annotation tools how far are we today?" in *Fourth International Conference on Language Resources and Evaluation (LREC) 2004*, vol. 1, Lisbon, Portugal, May 2004, pp. 197–202.
- [8] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Interspeech 2005*, September 2005, pp. 1517–1520.
- [9] C. Draxler, "Webtranscribe an extensible web-based speech annotation framework," in *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005, vol. 3658, pp. 61–68.
- [10] A. Pittermann and J. Pittermann, "Getting bored with HTK? using HMMs for emotion recognition," in *8th International Conference on Signal Processing (ICSP)*, Guilin, China, November 2006.
- [11] P. Boersma, "Praat, a system for doing phonetics by computer," *Glott International*, vol. 5, no. 9/10, pp. 341–345, 2001.