UNIVERSITY OF TARTU
DEPARTMENT OF ENGLISH STUDIES

Calculating the Error Percentage of an Automated Part-of-Speech Tagger when Analyzing Estonian Learner English – An Empirical Analysis

MA thesis

Aare Undo
SUPERVISOR: Lect. Jane Klavan

TARTU
2018

ABSTRACT

Assigning parts of speech tags to words is as old as linguistics, but computers have become powerful enough to automate this process only in the recent decades, so automated part-of-speech tagging is a relatively new field of study. Complex algorithms have been developed and improved over the decades. This thesis puts one of the most elaborate part-of-speech taggers to the test on Estonian learner English corpora. The background to the study is compiling the Tartu Corpus of Estonian Learner English (TCELE) at the Department of English Studies at the University of Tartu. The current status of the corpus is roughly 25,000 words (127 written essays) and 11 transcribed interviews (~100 minutes in total). The aim of the study is to compare the error rate of automatic tagging of TCELE to the error rates found for automatic tagging of native English corpora (e.g. the BNC) and other learner corpora.

The first part of this thesis introduces the reader to the compilation, annotation and retrieval of corpora, additionally, provides an overview of part of speech tagging, what are some of the more frequently used tag sets (or tagsets) and taggers available and explains the methods, approaches and degrees of success, using either rule-based methods or statistical approaches, or a combination of the two. The middle part of the the thesis is concerned with previous research and provides an overview of the following aspects in relation to automatically tagging learner corpus data, such as: What has previously been done? What were the main result? Which tagset and tagger were chosen?

The empirical part of this thesis is about automatic tagging of learner corpus texts and manual analysis and comparison of the findings. Both written and spoken corpora were automatically tagged using python scripts and Natural Language Toolkit (NLTK) library, a select number of essays (written text corpora) and interviews (spoken text corpora) were manually analysed in order to ascertain whether the words have been tagged correctly, then error rate was calculated. The empirical analysis shows that the error rates range from 0% (Essay 2) to 2.65% (Essay 3), with the average error rate being around 1.06%. For the spoken subcorpus of TCELE, the error rate was considerably higher – 23.14%.

The errors were analysed, problematic scenarios were presented, additionally, error rate was compared to other taggers used for tagging learner texts from different learner corpora of English. This thesis also looked into further research opportunities and provided a fairly detailed plan on how to make use of the annotated corpora, including a proof-of-concept level interactive corpus client. The main contribution of the study is the finding that based on the five essays analysed for the thesis, the average error rate is acceptable for researchers to trust the automatic tagging of the written subcorpus of TCELE in the current format. For spoken language, additional manual tagging is required. As a result of the study carried out for the purposes of this MA thesis, the TCELE corpus has been automatically tagged to allow future researchers easy retrieval of the necessary information.

Table of Contents

LIST OF ABBREVIATIONS

API – Application Programming Interface

BNC – British National Corpus

CLAWS – Constituent Likelihood Automatic Word-tagging System

FLT – Foreign Language Teaching

ICLE – International Corpus of Learner English

JSON – JavaScript Object Notation

NLP – Natural Language Processing

NLTK – Natural Language Toolkit

ORM – Object Relational Marking

POS – Part-of-Speech

SLA – Second Language Acquisition

TCELE – Tartu Corpus of Estonian Learner English

TOSCA-ICLE – Tools for Syntactic Corpus Analysis – International Corpus of Learner English

Introduction

Part of speech tagging is as old as linguistics, but computers have become powerful enough to automate this process only in the recent decades, so automated part-of-speech tagging is a relatively new field of study. Complex algorithms have been developed and improved over the decades. This thesis will put one of the most elaborate part-of-speech taggers to the test on the corpus of Estonian learner English compiled at the University of Tartu (TCELE). The main aim of the thesis is to ascertain whether the accuracy percentage is high enough for researchers and students working with TCELE to benefit from the automatically tagged texts.

The main objective for this thesis is to determine the accuracy rate based on NLTK (Natural Language ToolKit), an automatic part-of-speech tagger, and, if tagging accuracy is deemed high enough, to tag the entire TCELE corpus using NLTK . The research question the present thesis considers is the following: POS-taggers are usually designed for written native language; to what extent is it possible to use state-of-the-art automated part-of-speech taggers for analysing Estonian Learner English texts?

In order to allow for (future) researchers to easily retrieve necessary information, it is important to have an annotated corpus. The most common form of corpus annotation is part-of-speech (POS) tagging: a process where a label (tag) is assigned to each word in the text representing its major word class. TCELE, in its current a format, has not yet been annotated. This thesis will provide an overview of the quality of Estonian learner English included in TCELE and will help in the compilation process of TCELE, which at the moment is a work in progress. Additionally, the thesis provides some possibilities for future work in the field of automatic tagging of learner corpora.

In order to provide an answer to the research question, part of the TCELE corpus is automatically tagged. To validate the accuracy of the automatic tagging, manual tagging of the same sample of texts is carried out. The present thesis is interested in the error and success rates of manually tagging learner English texts. When the interest lies in comparing the texts produced by native and non-native speakers and additionally how well or badly an automatic tagger has performed, there are three possible scenarios. This thesis will, among other things, test which of the following scenarios is correct:

1. The success rate of the automated part-of-speech tagger is higher for native language, because there are errors/ innovations in learner language.
2. The success rate of the automated part-of-speech tagger for learner language is on a par with native language.
3. The success rate of the automated part-of-speech tagger is higher for learner language, because the learners' language is structurally less complex.

According to Manning (2011) the current part-of-speech taggers work very reliably on native speaker language with per-token accuracies of slightly over 97%. However, Manning (2011: 171) also points out that the "story is not quite so rosy" when researchers look at the rate of getting whole sentences right or when there are differences in topic, epoch, or writing style between the training data and operational data. He (Manning 2011: 171) claims that "a single bad mistake in a sentence can greatly throw off the usefulness of a tagger to downstream tasks such as dependency parsing”. In case of learner language, such mistakes are easy to occur and the training data for the automatic tagging (usually based on native speaker language) is different from the operational data (learner language).

The first part of this thesis introduces the reader to the compilation, annotation and retrieval of corpora, additionally, provides an overview of part of speech tagging, what are tag sets (or tagsets) and taggers. And finally, explains the methods, approaches and degrees of success, using either rule-based methods or statistical approaches, or a combination of the two. The middle part of the the thesis is concerned with previous research and provides an overview of the following aspects in relation to automatically tagging learner corpus data:

- What has been done previously in terms of tagging learner corpus data?
- How was it done?
- What were the main results?
- Which tagger was used for this particular corpora?
- Which tagset was chosen?
- How relevant is the choice of tagset and tagger relevant for the purposes of the present study?

The empirical part of this thesis will be about automatic tagging of TCELE learner corpus texts and manual analysis and comparison of the findings. After a corpus has been collected, compiled and marked up, comes the stage of annotation. Annotation can be applied using manual or automatic methods. This thesis will use a combination of the two on the first five essays in order to calculate the error rate. First, both written and spoken corpora are automatically tagged using python scripts and Natural Language Toolkit (NLTK) library. It is important to stress that all of the scripts used in the thesis are accessible via GitHub under the account of the author (https://github.com/Nikituh/scripts/tree/master/nltk) in order to allow for future researchers working on TCELE and other corpora to be able to use these scripts. Second, a select number of essays (written text corpora) and interviews (spoken text corpora) will be manually analysed, meaning that a human will go over the tagged words and verify (to the best of their ability) whether the words have been tagged correctly, then calculate the error percentage.

Finally, the thesis will analyse the errors, a descriptions is provided about the problematic scenarios and how to solve them; additionally, error rates are compared and general conclusions drawn, some further research opportunities will be highlighted and a fairly detailed plan on how to make use of the annotated corpora is provided.

1. PART-OF-SPEECH TAGGING OF LEARNER CORPORA

The first chapter of the thesis aims to give a short overview of the following main topics: the overall characteristics of learner corpora, the process of annotating a corpus, what is part-of-speech tagging and what are the various tagsets proposed in the literature and which are some of the more prominent automatic taggers used within the field. The aim of the chapter is to give background to the relevant literature that is essential for the empirical part of the thesis, where automated part-of-speech tagging is compared to manual tagging of learner English texts.

1.1. What are learner corpora?

Since the present thesis addresses the topic of automatically tagging learner corpus texts, it is important to, first of all, discuss what types of corpora do we consider to be learner corpora. Learner corpora are electronic collections of language data produced by L2 learners, i.e. second or foreign language learners. This relatively new resource is of great relevance for both second language acquisition (SLA) research and foreign language teaching (FLT). One of the main characteristics of learner corpus research is that it makes full use of corpus linguistic methods and tools to understand the process of language acquisition, describe L2 learner language varieties and design pedagogical tools that target learners' attested difficulties. The first learner corpus collections, which date back to the 1980s, only targeted English data. Since then, the field has expanded considerably and now includes learner data in a large number of languages. (Granger 2012). The aim of this thesis is to contribute to the field of learner corpus research by looking at Estonian learner English. The research area of Estonian learner English is currently largely underexplored save for a few studies, e.g. Tammiste (2016), Daniel (2015), Merilaine (2015), and Kirsimäe (2017).

As Granger (2012) explains in her paper, learner corpus data have the following distinguishing characteristics:

1. They are in electronic format
2. They have been compiled on the basis of strict design criteria, pertaining to the learner (age, mother tongue etc)
3. They contain continuous discourse rather than decontextualized words, phrases, sentences
4. They include data of the most open-ended type, ranging from fully natural (learners' communications with native speakers as they go about their normal business) to semi-natural (resulting from pedagogical tasks)

The fourth characteristic is the fuzziest, as there are many degrees of "naturalness". Nesselhauf (2004: 128) distinguishes between prototypical learner corpora, which display all the defining characteristics of learner corpora (e.g. argumentative essays or informal interviews), from peripheral learner corpora, which only partly fill these criteria (e.g. summaries or picture descriptions).

Learner corpora can be categorized along several dimensions, arguably, the most important are the following three:

1. The scope of the data collection
2. The time(s) of the data collection
3. The medium of the language data

The background to the study is compiling the Tartu Corpus of Estonian Learner English (TCELE) at the Department of English Studies at the University of Tartu. The current status of the corpus is roughly 25,000 words (127 written essays) and 11 transcribed interviews (~100 minutes in total). All of these will be annotated by NLTK, but only a select few will be manually analysed. If TCELE is to be considered from the perspective of the learner corpus criteria highlighted above, it can be concluded that in its current state, it is in an electronic format, it is compiled on the basis of the design criteria that it should be texts and speech produced by native or bilingual speakers of Estonian, it encompasses contextualized text and speech in the form of essays and interviews, and finally, as to the naturalness of the corpus, it tends to tilt towards the unnatural end of the continuum since it is essays produced as part of an entry exam and the interviews are semi-structured interviews. It is hoped that future work on TCELE will include other types of texts and speech for which the three first design criteria are the same, but the naturalness of which is towards the other end of the continuum.

1.2. Annotation

The following subsection gives an overview of annotation and what it means in the context of the present thesis. Kennedy (1998: 70) states that there are three stages to corpus compilation: “corpus design, text collection or capture and text encoding or markup”. Rayson (2015) talks about the following stages: compilation, annotation and retrieval. This paper will focus on the annotation stage. Part of the compilation for TCELE has already been done. Annotation is now the next crucial step to enable the retrieval stage that follows annotation.

For a successful corpus-based or corpus-driven research one either needs access to an already existing corpus or in case there is not a necessary corpus available, create your own corpus. Creating a machine-readable corpus can be very time consuming, as accuracy of transcripts and scans is a primary consideration (Rayson 2015). Once the corpus has been compiled and annotated, the retrieval process begins. Essentially, retrieval is a process where an annotated corpus is placed into a format where researchers can draw information about the data and subsequently draw conclusions based on the data. In other words, it would be highly desirable to develop a search engine on a website for the compiled corpus. Unfortunately, no search engine currently exists for TCELE and since building one requires both extensive skills and finances, it falls out of the scope of the present thesis and remains a challenge for the future. Still, as a possible future endeavour, a pilot version for retrieving the part-of-speech tagged TCELE corpus is provided at the end of the thesis, but it is currently in an experimental stage.

Annotation can take many forms: morphological, lexical, syntax, semantic, pragmatic, stylistic or discoursal. The present thesis focuses on morphological, part-of-speech tagging. Leech (2005) defines annotation as the practice of adding interpretative linguistic information to a corpus text. In the context of the present thesis, the added interpretative linguistic information is a specific part of speech tag that is added to each unit of a text. This way, it will be possible for researchers to later retrieve all the necessary data based on the linguistic information they are looking for. For example, should the researcher be interested, he or she can retrieve all the instances of prepositions since they have been added a tag that identifies them from the other parts of speech. One can also look for a specific word used as a specific part of speech, e.g. the use of *result* as a noun or a verb. In other words, having an annotated corpus is highly beneficial for research purposes.

Two important considerations need to be made here. First of all, in the era of big data, a researcher wants to get access to a very big corpus and wants it to be annotated. For this, automatic part of speech tagging can be used. At the same time, the researcher also wants to retrieve clean data and for this purpose, the automated tagging needs to be manually checked or at least, the researcher should know what is the percentage of error and how much additional manual checking he or she needs to do. Another upside to using manual tagging is that according to Rayson (2015: 6) “if the text is annotated or corrected by hand then this could form the basis of a training corpus for an automatic tagging system which can then learn from the human annotators in order to attempt to replicate their coding later on larger amounts of data”.

This thesis will attempt annotation using Natural Language Toolkit, a library (a collection of programs and software packages made generally available) that a software developer uses, however, there are various third-party tools available. Dexter is one such tool (http://dexter.sourceforge.net/). It is a little java program to interactively or semi- automatically extract data from scanned graphs, meaning that it extracts the text from scans and automatically annotates it in order to verify the validity of the scan. eMargin (https://emargin.bcu.ac.uk/) is a collaborative textual annotation tool, meaning that that it is used to simplify manual tagging. A trial version of CLAWS (Constituent Likelihood Automatic Word-tagging System) tagging client is freely available on the website of the University Centre for Computer Corpus Research on Language.

However, the main drawbacks of such methods of annotation are time and format. Whenever using an existing piece of software (be it online or offline), the user relies on the input and output format of said existing software. As previously mentioned, CLAWS has a tagger that is available online, but the server is slow and the word count is limited, tagging one ~200-word essay will take minutes. Therefore, it was decided not to use the CLAWS tagger, but to use the Natural Language Toolkit instead. A more detailed overview of this toolkit will come later.

1.3. Part-Of-Speech Tagging

1.3.1. What is part-of-speech tagging?

Part-of-speech-tagging (or POS tagging, PoS tagging, POST) is a branch of corpus linguistics. It is also called grammatical tagging or word-category disambiguation, it is the process of marking words in a text with its corresponding part of speech. The process is very complex when tagging complex parts of speech, such as entire essays produced by non-native speakers, but a simplified form of it is taught to even school children: the identification of words as nouns, verbs, adjectives etc. Once, the tagging of text was performed by hand, but now POS tagging is done in the context of computational linguistics, using algorithms which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags.

A key element of POS-tagging is tokenisation. Tokenisation is the process during which each unit for tagging is identified and separated from its surrounding tokens. This is usually a very straight-forward process of identifying orthographic words, but is complicated by punctuation marks, abbreviations and cliticised forms like the apostrophe ’s in English. Further complications for the English language arise in the case of hyphenated forms and compounds that are written as more than one orthographic unit. (van Rooy & Schäfer 2002: 327). These problems are evident in the present work as well. Tokenisation is particularly challenging for transcribed spoken text.

1.3.2. Methods, approaches and degrees of success

There are very many taggers available with varying degrees of success (cf. Schmid 1994, Garside et al. 1997, Müller & Strube 2006, Rayson 2015). Rayson (2015) discusses at least two approaches: **rule-based methods** and **statistical approaches.** The most successful taggers employ a combination of the two (Rayson 2015: 39). Rule-based methods rely on large manually constructed knowledge-bases encoding linguistic information such as the possible POS tags that a word or suffix may take and templates giving contexts where specific POS tags are ruled in or out (Rayson 2015). E. Brill's tagger, one of the first and most widely used English POS-taggers, employs rule-based algorithms.

Statistical approaches draw their information from large corpora and use probabilities to calculate which POS tag is most likely in a given context. The most successful taggers employ a combination of the two kinds to provide robust results across multiple types of text, e.g. CLAWS11 (Rayson 2015). Good automatic taggers, generally, have a success rate greater than 90%, but this is highly language-specific and also depends on the type of text. As pointed out by Manning (2011) “current part-of-speech taggers work rapidly and reliably, with per-token accuracies of slightly over 97%”.

1.3.3. Taggers

There are numerous part-of-speech taggers available, the most prominent are the following:

1. Stanford Log-linear Part-Of-Speech Tagger
2. OpenNLP
3. CLAWS part-of-speech tagger for English
4. Natural Language Toolkit (NLTK)

Stanford Log-linear Part-Of-Speech Tagger was originally written by Kristina Toutanova. Since that time, Dan Klein, Christopher Manning, William Morgan, Anna Rafferty, Michel Galley, and John Bauer have improved its speed, performance, usability, and support for other languages. The English tagger uses the Penn Treebank tag set. Penn Treebank tag set is a tag set consisting of 36 part-of-speech tags created by the The University of Pennsylvania. The basic download is a 24 MB zipped file with support for tagging English. The full download is a 124 MB zipped file, which includes additional English models and trained models for Arabic, Chinese, French, Spanish, and German. In both cases most of the file size is due to the trained model files (The Stanford Natural Language Processing Group website's POS tagger section). Stanford Log-linear Part-Of-Speech Tagger employs a combination of rule-based and statistical approaches.

The **Apache OpenNLP** library is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. These tasks are usually required to build more advanced text processing services. OpenNLP also included maximum entropy and perceptron-based (the simplest form of artificial neural network) machine learning (Apache OpenNLP documentation). Apache OpenNLP employs a combination of rule-based and statistical approaches.

Manning and Schutzhe (2001) offer a good explanation of maximum entropy. Basically it involves a framework for putting together information from many different sources for classification. The sources can be fairly heterogeneous. The classification problem boils down to deciding based on a (potentially large) number of features which features are important for classification. According to Manning and Schutzhe (2001: 589), "these features can be quite complex and allow the experimenter to make use of prior knowledge about what types of informations are expected to be important for classification". Constraints on the model are based on each of these features and the maximum entropy model is "the model with with the maximum entropy of all the models that satisfy the constraints" (Manning and Schutzhe 2001: 589). The basic idea is that they try to avoid going beyond the data.

If a model with less entropy is chosen, information constraints will be added to the model that are not validated by the empirical evidence available. Thus, maximum entropy model is motivated by the desire to keep as much uncertainty as possible (Manning and Schutzhe 2001: 589).

The goal of the OpenNLP project will be to create a mature toolkit for the above mentioned tasks. An additional goal is to provide a large number of pre-built models for a variety of languages, as well as the annotated text resources that those models are derived from.

As explained on the CLAWS web page of the University Centre for Computer Corpus Research on Language, the tagger has consistently achieved 96-97% accuracy for tagging native speaker texts (the precise degree of accuracy varies according to the type of text). Judged in terms of major categories, the system has an error-rate of only 1.5%, with c.3.3% ambiguities unresolved, within the BNC (British National Corpus).

Table 1 presents the error and ambiguity rates for the BNC. The size of the test corpora is 50,000 words, the error rate for written texts is 1.14% (Table 1) and the error rate for spoken texts is 1.17% (Table1). The University Center also calculated the error percentage after eliminating ambiguities (in the current context, "ambiguity" means making sentences more comprehensible, eliminating possible multiple-interpretations), the error rate actually rises to 2.01% (Table 2) for written texts and 1.92% (Table 2) for spoken texts. (University Centre for Computer Corpus Research on Language's BNC page). The potential reasons for such a rise after removing ambiguities will be discussed in greater detail in the discussion section. The BNC manual does not discuss this in detail.

Table 1. Error and ambiguity rate (BNC Manual)

	Sample tag count	Ambiguity rate	Error rate
Written texts	45,000	3.83%	1.14%
Spoken texts	5,000	3.00%	1.17%
All texts	50,000	3.75%	1.15%

Table 2. Error rate after removing ambiguity

	Sample tag count	Error rate
Written texts	45,000	2.01%
Spoken texts	5,000	1.92%
All texts	50,000	2.00%

Several tagsets have been used in CLAWS over the years. The CLAWS1 tagset has 132 basic word tags, many of them identical in form and application to Brown Corpus tags. A revision of CLAWS at Lancaster in 1983-1986 resulted in a new, much revised, tagset of 166 word tags, known as the ‘CLAWS2 tagset’. The tagset for the BNC (C5 tagset) has just over 60 tags. This tagset was kept small because it was designed for handling much larger quantities of data than were dealt with up to that point (University Centre for Computer Corpus Research on Language CLAWS page). CLAWS employs a combination of rule-based and statistical approaches.

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Most importantly, NLTK is a free, open source, community-driven project (NLTK Documentation). Natural Language Toolkit employs a combination of rule-based and statistical approaches. For this study, I determined that NLTK is the most appropriate part-of-speech tagger, because it is a leading platform based on Python and provides over 50 corpora and lexical resources as its sources. Plus, it is free and open source.

1.3.4. Tagsets

A tagset is, essentially, the number of different linguistic tags a tagger has to choose from. A smaller tagset will provide results with higher accuracy, because the algorithm has fewer options to choose from, but this also has obvious drawbacks. Different taggers have different tagsets. The Brill-tagger consists of 36 tags. The CLAWS7 tagset consists of 137 tags, excluding punctuation tags. TOSCA-ICLE (Tools for Syntactic Corpus Analysis and International Corpus of Learner English) consists of 220 different tags. The tagger used in this paper (NLTK) uses the Penn Treebank Tagset, the same as the Brill-tagger. So, in theory, NLTK and the

Brill-tagger should yield similar results, however, the internal corpora they use are different, so minor differences in tagging are likely to occur.

While TOSCA-ICLE has the highest number of different tags, CLAWS7 uses a more advanced system for tagging adverbs**.** The CLAWS7 tagset has separate tags for the various forms of the verbs ‘do’, ‘be’ and ‘have’, but does not distinguish the auxiliary and main verb functions of these three verbs. TOSCA-ICLE, on the other hand, makes this distinction. As an example of how specific various tagsets can be in their distinction of different levels of tags, one can have a look at the TOSCA-ICLE tagset which has the following 32 tags for pronouns:

1. PRON(antit)
2. PRON(antit,procl)
3. PRON(ass)
4. PRON(cleft)
5. PRON(cleft,procl)
6. PRON(dem,number)
7. PRON(dem,plu)
8. PRON(dem,sing)
9. PRON(exclam)
10. PRON(inter)
11. PRON(inter,poss)
12. PRON(neg)
13. PRON(nonass)
14. PRON(nomposs,number)
15. PRON(nomposs,plu)
16. PRON(nomposs,sing)
17. PRON(one)
18. PRON(pers,number)
19. PRON(pers,plu)
20. PRON(pers,plu,encl)
21. PRON(pers,sing)
22. PRON(pers,sing,procl)
23. PRON(poss,number)
24. PRON(poss,plu)
25. PRON(poss,sing)
26. PRON(quant)
27. PRON(recip)
28. PRON(rel)
29. PRON(rel,poss)
30. PRON(self,plu)
31. PRON(self,sing)
32. PRON(such)
33. PRON(univ

The more refined a tagset, the greater the risk of making errors with smaller category distinctions. When evaluating performance, it is therefore essential to make provision for the effect of tagsets on the results (Van Halteren 1999), and since this is an empirical study about turning learner language corpora into an interactive corpus (and its use is to find the types of errors learners make), it is important that the tagger's error rate be as low as possible. Therefore, it was decided that it is not necessary to differentiate between 33 different pronouns.

The choice of tagger and tagset largely depends on the type of research. Neither is it necessary to differentiate between the 16 types of adverbs CLAWS7 has:

1. **RA** adverb, after nominal head (e.g. else, galore)
2. **REX** adverb introducing appositional constructions (namely, e.g.)
3. **RG** degree adverb (very, so, too)
4. **RGQ** wh- degree adverb (how)
5. **RGQV** wh-ever degree adverb (however)
6. **RGR** comparative degree adverb (more, less)
7. **RGT** superlative degree adverb (most, least)
8. **RL** locative adverb (e.g. alongside, forward)
9. **RP** prep. adverb, particle (e.g about, in)
10. **RPK** prep. adv., catenative (about in be about to)
11. **RR** general adverb
12. **RRQ** wh- general adverb (where, when, why, how)
13. **RRQV** wh-ever general adverb (wherever, whenever)
14. **RRR** comparative general adverb (e.g. better, longer)
15. **RRT** superlative general adverb (e.g. best, longest)
16. **RT** quasi-nominal adverb of time (e.g. now, tomorrow)

As this thesis is concerned with general analysis of learner English, Penn-Treebank Tagset is the best choice as it is the most general-purpose tagset available. The following section provides details about the Penn-Treebank tagset.

1.4. Penn-Treebank Tagset

By default, NLTK uses 36 part-of-speech tags, as defined by Penn Treebank Project. For this thesis, I relied on the aforementioned default tags (since these were deemed sufficient from the perspective of the aims of the thesis). Words based on eight parts of speech: the verb (VB), the noun (NN), the pronoun (PR+DT), the adjective (JJ), the adverb (RB), the preposition (IN), the conjunction (CC), and the interjection (UH). The tags are as follows:

1.	**CC**	Coordinating conjunction
2.	**CD**	Cardinal number
3.	**DT**	Determiner
4.	**EX**	Existential there
5.	**FW**	Foreign word
6.	**IN**	Preposition/subordinating conjunction
7.	**JJ**	Adjective
8.	**JJR**	Adjective, comparative
9.	**JJS**	Adjective, superlative
10.	**LS**	List item marker
11.	**MD**	Modal
12.	**NN**	Noun, singular or mass
13.	**NNS**	Noun, plural
14.	**NNP**	Proper noun, singular
15.	**NNPS**	Proper noun, plural
16.	**PDT**	Predeterminer
17.	**POS**	Possessive ending
18.	**PRP**	Personal pronoun
19.	**PRP$**	Possessive pronoun
20.	**RB**	Adverb
21.	**RBR**	Adverb, comparative
22.	**RBS**	Adverb, superlative
23.	**RP**	Particle
24.	**SYM**	Symbol
25.	**TO**	to
26.	**UH**	Interjection
27.	**VB**	Verb, base form
28.	**VBD**	Verb, past tense
29.	**VBG**	Verb, gerund or present participle
30.	**VBN**	Verb, past participle
31.	**VBP**	Verb, non-3rd person singular present
32.	**VBZ**	Verb, 3rd person singular present
33.	**WDT**	Wh-determiner
34.	**WP**	Wh-pronoun
35.	**WP$**	Possessive wh-pronoun
36.	**WRB**	Wh-adverb

The inclusion of chunk tags (assigned to groups of words that belong together, i.e. phrases: noun phrase, verb phrase) can be beneficial, but not essential. It could be beneficial because it allows for more complex constructions to be tagged as chunks, as there is otherwise danger they might be mis-tagged, however, it is not essential, as this thesis is concerned with tagging words, not chunks, and it is expected that there are few such cases in the learner text that are analysed in the present thesis. Examples of the 36 parts of speech found in TCELE are:

1. **CC**: and, but
2. **CD**: one
3. **DT**: the, a
4. **EX**: there
5. **FW**: over (incorrect tag)
6. **IN**: along, in
7. **JJ**: sad, native
8. **JJR**: more
9. **JJS**: most
10. **LS**: –
11. **MD**: would
12. **NN**: world
13. **NNS**: nations
14. **NNP**: English
15. **NNPS**: Russians
16. **PDT**: all
17. **POS**: ‘s
18. **PRP**: You
19. **PRP$**: their
20. **RB**: always
21. **RBR**: more
22. **RBS**: most
23. **RP**: (point) out
24. **SYM**: –
25. **TO**: to
26. **UH**: –
27. **VB**: be
28. **VBD**: won
29. **VBG**: being
30. **VBN**: used
31. **VBP**: think
32. **VBZ**: is
33. **WDT**: which, that (incorrect tag)
34. **WP**: what
35. **WP$**: whose
36. **WRB**: when

As can be seen by the use of the en dash, LS, SYM and UH are not represented in the automatically tagged TCELE corpus.

1.5. Existing research on tagging non-native English

Automatic part-of-speech tagging has been the subject of research for decades. There is much to draw from studies based on tagging texts produced by native speakers. However, there are only a few studies that focus on automatic part-of-speech tagging of non-native English. A recent comprehensive overview of the issues related to the field and the major studies conducted can be found in van

Rooy (2015). Van Rooy (2015: 103) stresses that the annotation of learner corpora considerably increases the value of the data for research - it makes it possible to extract data from learner corpora that would otherwise not be accessible. As one of the major concerns for future research, van Rooy (2015: 103) mentions the need in the field for more publicly available annotated data, especially data that has been manually checked, as is the case with the present study. However, since putting together learner language corpora and the annotation process take a lot of manual labor and since the material in the corpus may be of sensitive value, research units are understandably careful in publishing such data. The same considerations hold for the present study on TCELE as well.

The present thesis draws a lot of inspiration from two specific studies on automatically tagging learner corpus data: the studies conducted by de Haan (2000) and van Rooy and Schäfer (2002) and hence these two will be discussed in detail. It is important to study existing research and determine what and how has been studied already, e.g. whether it was just learner language that was of interest, or a comparison with native texts. Furthermore, it is also necessary to determine what kind of conclusions have been drawn, and whether my research correlates with it. In what follows, a short overview of the previous work on tagging non-native English is given.

1.5.1. Tagging non-native English with the TOSCA-ICLE tagger

The TOSCA (Tools for Syntactic Corpus Analysis) working group at Nijmegen University developed a computerised method for interactive syntactic analysis of unprepared text material. This method yields a data base of syntactically analysed material in the form of trees (van Den Heuvel 1988). The TOSCA-ICLE Tagging Unit (TU) has been in use for some time now to tag (part of) the material in several research centres participating in the ICLE (International Corpus of Learner English) project. The TU comprises among other things an automatic tagger, and a tag selection program, which can be used to correct tagger output (the tagger has a success rate of approximately 95 per cent with Spanish written non-native material) (de Haan 2000).

Table 3 presents the examples of the various error types distinguished on the basis of Spanish ICLE material (de Haan 2000). De Haan (2000) examined a number of different types of learner errors in order to determine how best to correct POS-tagging errors. The author of this thesis will attempt to create a similar table for error rates in NLTK and Estonian learner English corpora.

Table 3. Examples of the various error types distinguished on the basis of Spanish ICLE Material (de Haan 2000 : 74)

error type	**examples**
obvious keyboard error	inly (for only), pepole (for people), reams (for dreams)
word class transfer	proud (for pride), creative (for creation), easily (for easy)
verb morphology error	became (for become), show (for shown), lives (for live)
grammatical error	much (for many), of (for on – as in dependent of) present (for to present), pretend (for pretending)
L1-lexis related	mean (for means), the poors (for the poor)
L1-morphology-related	criminality (for crime), differents (for different), ethic (for ethical)
L1-spelling-related	profesional or proffesional (for professional), posible (for possible), confort (for comfort)
L1-pronunciation-related	improve (for improved), baticano (for Vatican)
hypercorrection	anorexy (for anorexia)

1.5.2. Comparison of TOSCA-ICLE, CLAWS7 and Brill taggers

In order to decide which automatic tagging programs to use for the Tswana Learner English Corpus (TLEC), van Rooy and Schäfer (2002) evaluated the performance of three taggers on a small sample of the corpus (~ 2,000 words). They chose three taggers for their study: TOSCA-ICLE, Brill, and CLAWS (van Rooy and Schäfer 2002). One of the major conclusions of their study is that CLAWS is the most accurate of the three with accuracy of 98% and that learner errors significantly impact tagger accuracy (van Rooy and Schäfer 2002: 334).

Based on their empirical study, van Rooy and Schäfer (2002) regard CLAWS7 as “somewhat” of an industry standard, therefore an overview is also provided in this thesis. It is described in detail in many textbooks, and a number of important corpora have been tagged with CLAWS. Van Rooy and Schäfer (2002) selected Brill because it is purely rule-based and custom rules can be entered. An example of a custom rule would be: *The word “would” is always tagged as a modal if it is followed by “be”.* This example is very basic and obvious, real rules would be more complex, can contain several if-cases, etc. There can be hundreds of such custom rules, eliminating thousands of potential erroneous tags.

Van Rooy and Schäfer (2009) randomly selected five essays of roughly 400 words from the entire existing corpus, for a sample of just more than 2,000 words, or 1.25% of the current TLE (Treebank of Learner English) corpus. Three essays were tagged with all three taggers after which both of the authors jointly examined the tagger output and marked all incorrect tags with their corrections. One of the issues that they point out is the influence of spelling errors on the accuracy rate of automatic taggers. For their sample corpus, 78 spelling errors occurred out of 2,159 words, i.e. 36 spelling errors per 1,000 words (Van Rooy and Schäfer 2002: 331). Similarly, de Haan (2000: 70) has also identified spelling mistakes as one of the main sources of tagger errors in his work on the Czech, Dutch and Spanish Learner English Corpora. Van Rooy and Schäfer (2002) look into the matter by analysing the influence of spelling errors on tag errors. The findings of van Rooy and Schäfer (2002) are given in Table 4. The first column in Table 4 lists the categories of errors identified by van Rooy and Schäfer (2002), the second column lists the frequency of these errors and the remaining three columns indicate how well the three different taggers used in their research can handle the spelling errors.

Table 4. Influence of spelling errors on tag correctness (taken from van Rooy and Schäfer 2002: 332, Table 3)

Category	Total errors	TOSCA correct	Brill correct	CLAWS correct
Non-word errors	38	14	20	30
Real-word errors	14	1	5	5
Capitalisation	3	1	1	2
Space missing	10	0	0	0
Extra space	13	0	0	0
Total	78	16	26	37

Based on de Haan (2000: 71), van Rooy and Schäfer (2009) divide spelling errors into two broad categories - word errors (including spelling errors resulting in non-words, real words, capitalisation) and space errors (cases where two words are written as one word or where a single word is written as two words). Van Rooy and Schäfer (2002: 332) point out that errors with spacing always cause a tag error in all three taggers that they evaluated (cf. Table 4). As for word errors, the authors (Van Rooy and Schäfer 2002: 332) conclude that these can be handled in some cases, because the taggers employ a guessing module to assign tags to non-words and the various taggers have varying degrees of success in their guessing strategies. To determine the effect of spelling mistakes on tagger performance, van Rooy and Schäfer (2009) manually edited the corpus sample before re-tagging it. The results of their analysis are given in Table 5.

Table 5. Tagger performance after spelling correction (taken from van Rooy and Schäfer 2002: 333, Table 4).

Category	Total errors	TOSCA correct	Brill correct	CLAWS correct
Non-word errors	38	36	28	38
Real-word errors	14	12	13	13
Capitalisation	3	3	3	3
Space missing	10	4	2	10
Extra space	13	11	11	13
Total	78	66	57	77

As can be seen from Table 5, improvements are impressive. CLAWS assigned a correct tag to all but one of the corrected forms, while the other two also improved significantly. Van Rooy and Schäfer (2002: 33) point out that spelling is the cause of 18% of the tag errors in TOSCA and 13% in Brill, but 47% of the tag errors in

CLAWS. Thus, for CLAWS, removing spelling errors removes almost half of the tagger errors.

Since the aim of Van Rooy and Schäfer (2002) was to assess the usefulness of different taggers in tagging learner corpora, they present the results of tagging accuracy for both the raw text sample (i.e. unedited sample, given in Table 6) as well as for the edited sample (i.e. the version in which the spelling mistakes have been corrected, given in Table 7) for an overall comparison.

Table 6. Tagger accuracy on the raw sample (2159 tokens) (based on van Rooy and Schäfer 2000: 334, Table 6)

Tagger	TOSCA	Brill	CLAWS
Total errors	273	232	85
Accuracy	87%	89%	96%

Table 7. Tagger accuracy on the edited sample (2159 tokens) (based on van Rooy and Schäfer 2000: 334, Table 7)

Tagger	TOSCA	Brill	CLAWS
Total errors	223	201	45
Accuracy	90%	91%	98%

The conclusion drawn by van Rooy and Schäfer (2002) is that the majority of errors (two thirds) in CLAWS are due to learner errors, while a third to a quarter of errors in TOSCA and Brill can be attributed to learner errors. Van Rooy and Schäfer (2002: 334) concluded that for their sample of learner corpus texts, CLAWS is the most accurate with an accuracy of 98%. The implication for the present thesis is clearly to use CLAWS for tagging TCELE as well. However, as pointed out earlier CLAWS is not free and open source; in addition, the study by van Rooy and Schäfer (2002) clearly indicated that CLAWS does not perform well with spelling errors. For these and other reasons (e.g. being able to use the tagger with Python), the selected tagger for this study is NLTK. The next chapter presents the results of the empirical study conducted on a sample of TCELE texts to assess the usefulness of NLTK for automatic part-of-speech tagging of learner English.

2. AN EMPIRICAL STUDY OF AUTOMATIC TAGGING OF TCELE WITH NLTK

The second chapter of the thesis presents the empirical study of automatic part-of-speech tagging of TCELE in order to assess the usefulness of NLTK for these purposes. First, an overview of the NLTK library is given, followed by a detailed presentation of the process. All the relevant codes used in the thesis are made available in order to facilitate the same process to be applied once additional materials are added to TCELE. Hence, care is taken to explain all of the necessary steps and terminology used in the process since these may not be familiar to the reader. The second part of the empirical study involves a manual evaluation of the accuracy of the automatic tagging. For this purpose five essays are sampled from the total of 127 essays available. The empirical part of the thesis ends with the discussion of the types of errors made by the tagger.

2.1. Automatic tagging with NLTK

2.1.1. Library

NLTK is a Python-based library. Python-based software is convenient because it is easy to run on any major platform (Windows, Linux, OSX) and is a high level scripting language. “High level”, in this context, means that it is more human-readable (“Low level” languages mean closer to hardware that introduce a variety of additional complications) and, additionally, there is no compilation process (the process of transforming human-readable code into machine code. In order to properly tag texts, we first need to install NLTK's libraries and other auxiliary libraries required to parse input data. Total required download is approximately 4.0GB, NLTK library (of corpora, if you will) download requires approximately 3.5GB of disc space.

2.1.2. Parsing source texts

As the texts in the TCELE corpus are word (.doc, .docx) files, which contain additional metadata about fonts, sizes, colors, layout etc, (which, of course, makes sense, as it makes the data more human-readable) before we can even begin our POS tagging, we need to filter out such irrelevant tags. To achieve that, Python's `doc2txt` library was used (Sutherland 2018).

Programming, in general, is very modular. If some functions or algorithms have already been implemented somewhere, there is no need to rewrite them from scratch. As programmers are highly efficient, they reuse whatever they can. If someone wants to parse some text, for example, they write a piece of software that they most likely make freely available, so others can reuse their software (libraries) as they please.

As a final note, in automatic text analysis (and software development in general), such concepts as “pages”, “headers”, or “footers” (and the aforementioned fonts, sizes, colors, layouts, tables etc) do not exist. This is realized by the metadata provided when rendering the text to the end user. What is analysed is just plain text, line by line. It is possible to differentiate paragraphs by the amount of whitespaces used. Additionally, part-of-speech taggers do not output them in a coherent, readable format. To be able to write the parts of speech and tags into an excel or word document, additional formatting is required.

2.1.3. Tagging of written texts

Firstly, in addition to being a .doc file, the TCELE written text corpora contains page headers that are not defined as “headers” as per microsoft format, but as part of the text, so I had to filter those out. Luckily, each page header starts with `0114`, so I could just filter out all the lines that start with the said tag. These numbers are filtered out only for the automated tagging phase, of course, as they refer to the type of learner (sex, age etc.), which in turn can provide valuable insight into why a certain learner makes certain mistakes, or why the error rate of a certain essay may be much higher than the median.

Now that all relevant lines can be read, we can finally tokenize (separate into “tokens”, words, punctuation marks etc) the corpus. When it is separated, we apply the part-of-speech tagging algorithm. Now that the initial tagging is completed, we further need to transform and modify the output into a human-readable format. The initial output of a token is in the following format: `(u'Estonia', 'NNP')` (keep in mind that NLTK also tokenizes numbers and punctuation marks). This is what computer scientists call a tuple (a data structure consisting of multiple parts) and this specific tuple contains a unicode (a computing industry standard for the consistent encoding, representation, and handling of text) string and a regular string. From it we can extract `Estonia` and `NNP`, the latter of which we can use as a key to pull relevant data from the Penn Treebank Project and write it into an array (an array is a systematic arrangement of similar objects, usually in rows and columns). When the parsing is complete, it is written to a file, after which manual processing can begin.

2.1.4. Tagging of spoken texts

When analysing spoken texts, the first major question is: which of the following two approaches should be taken?

1. Separate the text into different speakers and analyze it one by one
2. Parse the text as a whole

For this thesis, it was decided to combine the texts of both speakers, e.g. sentence of speaker 1 is followed by sentence of speaker 2 as the sentences are erratic and often elliptical, it made more sense to combine the texts into one big chunk.

Before tagging can commence, it is first necessary to parse raw lines that are read by the Python script from a text document. The following steps need to be taken:

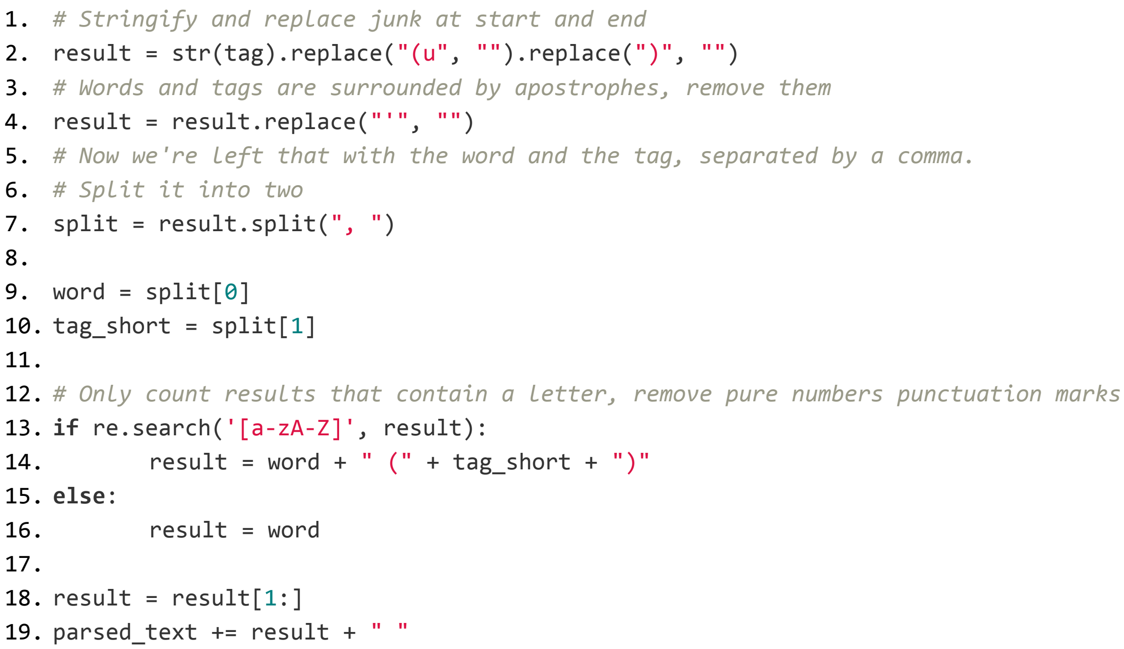
1. Remove the first three characters in the text, as they are just line numbers
2. Remove leading and trailing white-spaces
3. Remove all instances of **:** as they denote emphasis, intonation, breaks in pronunciation
4. Remove all tags inserted by the transcriber (<tagname> </tagname>), usually overlapping speech
5. Remove everything surrounded by parentheses and square brackets (these include unintelligible words)
6. Remove **=** characters
7. Remove **@** characters

8. Remove punctuation
9. Remove all speaker identifiers from lines

When the raw lines are parsed, tagging can commence. Here we follow the same pattern as we did for written texts. First, we tokenize the sentences into separate entities (words, numbers, punctuation marks), then we tag those entities. Then, we loop (a loop is a sequence of statements which is specified once but which may be carried out several times in succession) over our list of words.

The following Python snippet (presented in Figure 1) first removes `(u` and `)` (this logic is the same for both spoken and written texts, the **(u** represents a unicode string) characters from the words, then removes apostrophes (an alternative would be to extract the word and the tag from the tuple, but the current script stringifies the tuple first, a more *lucrative* approach). When we have successfully removed junk data, we split (this function splits a string object into an array of strings by separating the string into substrings, using a specified separator string to determine where to make each split) the strings (in computer programming, a string is traditionally a sequence of characters, either as a literal constant or as some kind of variable) by commas, as words and tags are formatted as: `"<word>, <tag>"`.

Now we have successfully formatted our corpora so that we have each word and its corresponding tag. We can format it however we want. For textual readability, the format I chose is: `word (tag) word (tag) word (tag)`. However, the script can be easily modified to fit different needs, e.g. tag checking is easier when each word is on a new line.



```
# Stringify and replace junk at start and end
result = str(tag).replace("(u", "").replace(")", "")
# Words and tags are surrounded by apostrophes, remove them
result = result.replace("'", "")
# Now we're left that with the word and the tag, separated by a comma.
# Split it into two
split = result.split(", ")

word = split[0]
tag_short = split[1]

# Only count results that contain a letter, remove pure numbers punctuation marks
if re.search('[a-zA-Z]', result):
        result = word + " (" + tag_short + ")"
else:
        result = word

result = result[1:]
parsed_text += result + " "
```

Figure 1. Script 1 used in the analysis

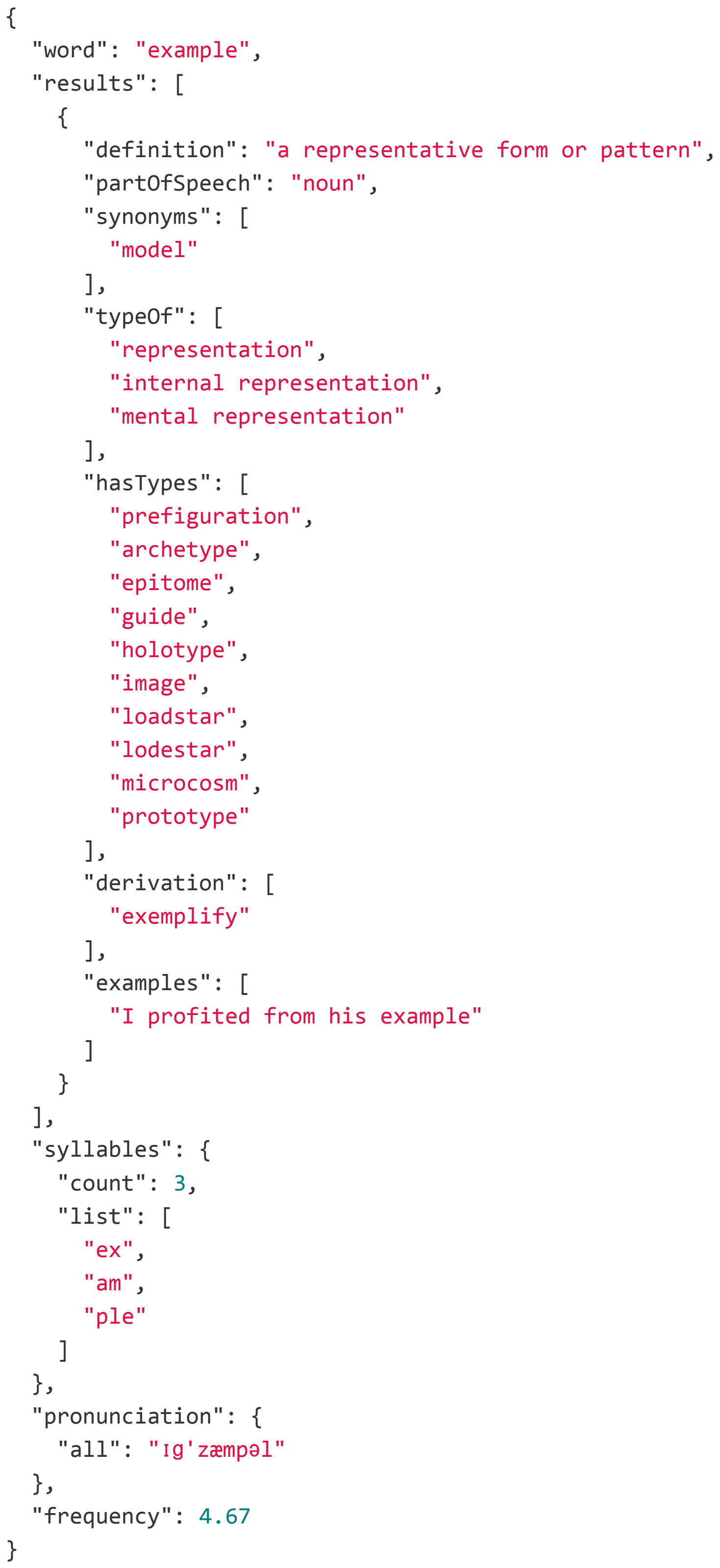
2.2. Manual analysis

After the essays have been tagged by NLTK, the next stage involves manually double checking the results in order to calculate the error percentage. Manual analysis is going to be done for five written essays and part of one spoken interview. The five manually analysed essays and part one interview are available in the appendices. For each essay, the gender and age of the author is specified in the original code produced by the compilers of TCELE. The automatically POS tagged essays are listed in Appendix 1. The total number of the words in the five essays is 982 - this constitutes the sample size for the manual checking of automatic tagging.

2.2.1. Words API

To simplify the process of manual tagging, the help of Words API service was used (https://www.wordsapi.com/). Words API is a dictionary for more than 150,000 words, additionally, it is a thesaurus, as it offers synonyms for words. Additionally, Words API includes hierarchical information, such as knowing that a hatchback is a type of car; a finger is a part of a hand; oxygen is a substance of water. Their entire library costs $629, but they offer limited free use. The free tier was sufficient for this thesis. This thesis uses Words API for its part-of-speech tagging capabilities. The service takes a word as an input and the output is all the possible part of the speech it can take the form of. Even though the service is not based on the Penn-Treebank Tagset, but rather base forms, it still tremendously simplified the process of manual tagging.

When querying the word *example* using the WordAPI service, the response is the following as can be seen in Figure 2 (the outcome has been shortened for brevity, the word *example* actually has six definitions). This format is known as JSON (JavaScript Object Notation) as is well-known in the developer community, as it provides a standard format a machine can easily analyze, yet a human can read as well.



```
{
  "word": "example",
  "results": [
    {
      "definition": "a representative form or pattern",
      "partOfSpeech": "noun",
      "synonyms": [
        "model"
      ],
      "typeOf": [
        "representation",
        "internal representation",
        "mental representation"
      ],
      "hasTypes": [
        "prefiguration",
        "archetype",
        "epitome",
        "guide",
        "holotype",
        "image",
        "loadstar",
        "lodestar",
        "microcosm",
        "prototype"
      ],
      "derivation": [
        "exemplify"
      ],
      "examples": [
        "I profited from his example"
      ]
    }
  ],
  "syllables": {
    "count": 3,
    "list": [
      "ex",
      "am",
      "ple"
    ]
  },
  "pronunciation": {
    "all": "ɪg'zæmpəl"
  },
  "frequency": 4.67
}
```

Figure 2. Shortened output of the WorAPI service for the word *example*

As one can see, in addition to other data, the response contains a list of objects that have the tag part of speech: `"partOfSpeech": "noun"`. This query was applied to every word in the analysed TCELE corpus texts and appended the possible types to every entry. The resulting lines have the following format: `of (IN) - preposition +`. In Penn- Treebank Tagset **IN** stands for preposition or subordinating conjunction. From this, one can easily distinguish that the *of* in question truly is a preposition. Of course, this process also had to be automated, as it would be too laborious to do it manually and would be as time-consuming as regular manual tagging. The following script as presented in Figure 3 was written to automate this process. Line numbers have been added to the original script in order to facilitate reference to the specific sections within the text.

```
1.  word_without_tag = parsed_word.split(" (")[0]
2.
3.  # Clean the word, as it can contain some junk data (example: . But)
4.  clean_word = word_without_tag.replace(",", "").replace(".", "")
5.  clean_word = clean_word.replace(":", "")
6.  clean_word = clean_word.replace(";", "")
7.  clean_word = clean_word.replace("(", "").replace(")", "").strip()
8.
9.  url = base_url + clean_word
10. request = urllib2.Request(url)
11.
12. for key, value in basic_headers.items():
13.         request.add_header(key, value)
14.
15. request.add_header('X-Mashape-Key', mashable_key)
16.
17. print("Requesting analysis for word: " + clean_word + " (" + str(counter) + "/" +
18. str(total) + ")")
19. counter += 1
20.
21. try:
22.         # Can throw: 404 Not Found
23.         # { "success":false,"message":"word not found" }
24.         response = urllib2.urlopen(request)
25. except:
26.         parsed_analyzed_words.append(parsed_word + separator)
27.         Continue
28.
29. result = response.read()
```

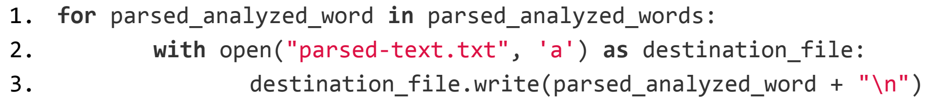
```
30. result_json = json.loads(result)
31.
32. # Add add the un-analyzed word to the list and continue to the next item
33. if "results" not in result_json:
34.         parsed_analyzed_words.append(parsed_word + separator)
35.         Continue
36.
37. result_json = result_json["results"]
38.
39. possible_tags = []
40.
41. for child in result_json:
42.         tag = child["partOfSpeech"]
43.         if not tag in possible_tags and tag != None:
44.                 possible_tags.append(tag)
45.
46. possible_tags = "/".join(possible_tags)
47.
48. parsed_analyzed_word = parsed_word + separator + possible_tags
49.
50. parsed_analyzed_words.append(parsed_analyzed_word)
```

Figure 3. Script 2 used in the analysis

Lines 4-7 clean the word of junk data, lines 10-30 make the actual request to Words API service (not relevant in the context of this thesis, except for line 26, where if Words API's database does not contain the word, a blank addition is made to the line). Lines 41-44 add all the possible tags to the line, the check for duplicates is on line 43. Line 46 joins the tags, separating them with slash (/) symbol. Line 50 adds the complete line to the array of lines that are written to a file. Each tagged word is formatted as: `Some (DT)`. So, in order to use the WordAPI service, the existing tag `(DT)` needs to be stripped (The method `strip()` returns a copy of the string in which all whitespace characters have been stripped from the beginning and the end of the string.), after that, strip away any unnecessary white space characters. Furthermore, tagged words can contain punctuation marks (e.g. `. For`), those need to be removed as well. The `clean word` section of the script removes any punctuation marks and braces from words.

As mentioned before, the word *example* has six different definitions, but all of them are noun, so it is only necessary to add one instance of the word “noun”. Which is great, because if NLTK tagged it as a noun, and wordsapi tagged it as a noun, then it is possible to make a reasonable assumption that the word is, in fact, a noun. It is also possible that a word has no definitions in wordapi.com corpus, a case that must be accounted for. The resulting line looks the following: `new (JJ) - adjective/adverb`. *new* is the word itself, (JJ) is NTLK’s tag for adjective, and adjective/adverb are the two possible parts of speech from wordsapi.com. It is possible to get thousands of such double-tagged words within seconds, when looping over all the words of the corpora.

The small snippet in Figure 4 goes over all entities (`parsed_analyzed_words` is an array of lines mentioned in the previous paragraph) and writes them to a file. `"\n"` is a new line character and it is added to each entry written to a file. This ensures that each word (and its accompanying tags) are written on to one line, and the following word to the next one.



```
for parsed_analyzed_word in parsed_analyzed_words:
        with open("parsed-text.txt", 'a') as destination_file:
                destination_file.write(parsed_analyzed_word + "\n")
```

Figure 4. Script 3 used in the analysis

As a final reference for manual analysis, The Free Dictionary by Farlex, was used in order to correctly check the tag of the words. When unsure of a word’s tag (and there were several complex situations), the examples provided on the site were consulted. If confusion persisted still, the Oxford English Dictionary was consulted (more on this in the *Linguistic problems* section of the thesis).

2.2.2. Error rate calculation

The script presented in Figure 5 was used to calculate NLTK's error rate.

```
1.  lines = []
2.
3.  total_count = 0;
4.  correct_tag_count = 0
5.
6.  def get_percentage(part, whole):
7.    return 100 * float(part)/float(whole)
8.
9.  with open("parsed-text-manually-analyzed.txt", 'r') as file:
10.
11.        lines = file.readlines()
12.        total_count = len(lines)
13.
14.        for line in lines:
15.               split = line.split(" ")
16.               manual_tag = split[len(split) - 1].strip()
17.
18.               if manual_tag == "+":
19.                      correct_tag_count += 1
20.               elif manual_tag == "-":
21.                      print("Incorrect tag: " + line)
22.               elif manual_tag == "#":
23.                      print("Nonexistant symbol: " + line)
24.               elif manual_tag == "!":
25.                      correct_tag_count += 1
26.               print("The developer should write a better parser: " + line)
27.
28. percent = get_percentage(correct_tag_count, total_count)
29. error_rate = 100 - percent
```

Figure 5. Script 4 used in the analysis

This script present in Figure 5 takes a text (.txt) document as its input, goes over the lines, reads the manual tag and, if correct (line 18), increments the correct tag count variable (line 19). When all lines are parsed, error rate is calculated and displayed.

The manual tag must be the final character of a line, space-separated. The input document expects the following format on each line of the input document. First the word, then NLTK's tag, then slash-separated wordapi's suggestions and finally the manual tag:

```
<word> (<tag>) - word/api/suggestions <manual-tagging-symbol>
```

Examples of the output of the manual tagging process:

1. turn (VB) - verb/noun +
2. into (IN) - preposition +
3. English-like (JJ) - +
4. mixtures (NNS) - noun +
5. . Other (JJ) - adjective +
6. positive (JJ) - adjective/noun +
7. sides (NNS) - noun/adjective/verb +

The following is the list of the manual tags used in the evaluation process:

1. **+** – Correct tag
2. **-** – Incorrect tag
3. **#** – Nonsense input
4. **!** – The tag itself was correct, the parser was incorrect

The nonsense input tag (#) is used when e.g. NLTK splits contractions into words and also attempts to tag the single quotation mark: \u2019 (unicode character for single quotation mark). Parser errors, tagged with an exclamation mark, include, for example, *today's*, that is a single word, but NLTK splits it into three different characters (today, \u2019, s) and tags all three, then the manual tag of the final *s* would be an exclamation mark.

2.2.3. Linguistic problems

There are certain words in English that can be categorised under several parts of speech. Take the following example from The Free Dictionary, presented in Figure 6:

an·y

(ĕn′ē)
adj.
1. One or some; no matter which: *Take any book you want. Do you have any information on ancient Romanarchitecture?*
2.
a. No matter how many or how few; some: *Are there any oranges left?*
b. No matter how much or how little: *Is there any milk left?*
3. Every: *Any dog likes meat.*
4. Exceeding normal limits, as in size or duration: *The patient cannot endure chemotherapy for any length of time.*
pron. (used with a sing. or pl. verb)
Any person or thing or any persons or things; anybody or anything: *We haven't any left. Any of the people behind thefront desk can help you.*
adv.
To any degree or extent; at all: *The patient didn't feel any better after the treatment.*

Figure 6. Definition of *any* as an adjective taken from The Free Dictionary

If we scroll down to definitions from another source, we see that it can also be a determiner, as presented in Figure 7.

any

(ˈɛnɪ)
determiner
1.
a. one, some, or several, as specified, no matter how much or many, what kind or quality, etc: *any cheese in thecupboard is yours*; *you may take any clothes you like*.
b. (*as pronoun; functioning as sing or plural*): *take any you like*.
2. (*usually used with a negative*)
a. even the smallest amount or even one: *I can't stand any noise*.
b. (*as pronoun; functioning as sing or plural*): *don't give her any*.
3. whatever or whichever; no matter what or which: *any dictionary will do*; *any time of day*.
4. an indefinite or unlimited amount or number (esp in the phrases **any amount** or **number**): *any number of friends*.

Figure 7. Definition of *any* as a determiner taken from The Free Dictionary

Thus “any” in the sentences ***Any*** *dog likes meat* and *any cheese in the cupboard* can be classified as either an adjective or a determiner depending on the system of word classes employed in different dictionaries. Here the question is of the concept of a word class in general and whether different dictionaries use different labels for one and the same part of speech. When comparing the automatic tagging to manual tagging we need to be careful with our judgements of the correctness of the automatic tagging output since it depends on the tagset used by the tagger. For these two particular examples of *any*, the expected automatic tag assigned by NLTK should be determiner [DT].

2.3. Error rates of the tagged texts from TCELE

2.3.1. Initial error rates of written texts

The initial error rate calculated after the first stage of manual evaluation for the first three essays was as follows:

1. Essay 1: NLTK error rate: 14.4%. Word count: 221
2. Essay 2: NLTK error rate: 7.3%. Word count: 123
3. Essay 3: NLTK error rate: 8.9%. Word count: 189

2.3.2. Severity of errors

Errors have different levels of severity. For example, there are four different types of pronouns in Penn-Treebank tagset and five different types of verb forms; these categories are given in Table 8.

Table 8. Types of pronouns and verb form types in the Penn-Treebank

Personal pronouns	Types of verb forms
1. Personal pronoun 2. Possessive pronoun 3. Wh-pronoun 4. Possessive wh-pronoun	1. Verb, base form 2. Verb, past tense 3. Verb, gerund or present participle 4. Verb, past participle 5. Verb, non-3rd person singular present 6. Verb, 3rd person singular present.

If NLTK tagger incorrectly tagged a specific type of pronoun or a specific type of verb form, the severity of the error is significantly smaller than if NLTK had tagged a verb as a noun or an adjective. However, the severity of this type of errors depends on the research questions of the researchers who will end up using the output of the automated error tagging. For example, for somebody interested in how well Estonian learners of English form the past tenses of English verbs, it would be very useful if the automatic tagger were able to assign correctly all of six different types of verb forms.

Questionable examples include “more”, which is sometimes tagged as a comparative adjective (JJR), sometimes as just an adjective (JJ) and “easier”, which is usually tagged as a comparative adjective, but not always. Additionally, Penn-Treebank tagset has a tag called “Existential there” (An existential clause is a clause that refers to the existence or presence of something. Examples in English include the sentences "There is a God" and "There are boys in the yard”). The existential there tag is either an adjective, pronoun or an adverb and, in the author's opinion, should be left out of the tagset, as it increases the error rate and complexity of manual analysis. David Crystal has noted that the *existential there* is entirely different from *there* used as a place adverb: "It has no locative meaning, as can be seen by the contrast: *There's a sheep over there*. The existential there carries no emphasis at all, whereas the adverb does: *There he is* (Rediscover Grammar, 2003).

Additionally, numbers, when presented in textual form as adjectives or nouns in sentences, are always tagged as “cardinal number”. It is difficult to argue which should be the correct tag in such a case. It is an adjective, but it is also, still, a cardinal number. The question of what should be the course of action in such a case is not straightforward.

As such, for the compilation of this study, I have decided to add the following exceptions to manual analysis:

1. If a word has been tagged as as a verb, but an incorrect type of verb form (the tense must be correct), the tagger has been correct
2. If superlative or comparative adjectives have been tagged as “just” adjectives, the tagger has been correct

3. If “there” has been tagged as an existential there, and it is a pronoun, the tagger has been correct
4. If Numbers, even when used as adjectives or nouns, are tagged as “cardinal number”, the tagger has been correct

No other exceptions have been made, e.g. when *when* is categorised as “just” an adverb, while the tag of wh-adverb exists.

2.3.3. Adjusted error rates of written texts

Following is the calculation of the error rates for the three essays sampled for the present study adjusted for the exceptions explained in the previous paragraph, the final error rate is as follows:

1. Essay 1: NLTK error rate: 0.45%. Word count: 221
2. Essay 2: NLTK error rate: 0.00%. Word count: 123
3. Essay 3: NLTK error rate: 2.65%. Word count: 189

Based on minor changes to the “algorithm” of manual analysis, the error rate becomes abysmal and this study takes a new turn. Automatic taggers suddenly become viable. Two further essays have been tagged and analyzed only based on this new set of adjusted rules.

4. Essay 4: NLTK error rate: 1.22%. Word count: 245
5. Essay 5: NLTK error rate: 0.98%. Word count: 204

For these five texts sampled from a set of 127 essays available in TCELE, a total of ~1000 words (982) was manually checked, which is a large enough sample size to draw conclusions from. However, one must bear in mind also the expectations allowed for the original automatic tagging. The combined error rate is **1.06%**, which is significantly lower than the error rates of TOSCA-ICLE or Brill, and even lower than that of CLAWS. However, it should be pointed out that there is considerable individual variation in the error rates across the five essays. For some essays the error rate is 0% (Essay 2, which is the shortest essay in the sample), but for some essays the error rate is above 2% (2.65% for Essay 3, which is of medium length). It cannot be concluded that length is the only determining factor in the correctness of the automatic tagging of an essay.

2.4. Detailed analysis of the sample of texts from TCELE

2.4.1. In depth analysis of the five written essays

Following is a detailed analysis of some of the most prominent cases where the automatic tagging with NLTK and the manual tagging were different. First, however, a few instances are given where it is surprising that the automatic tagger has been successful. For example, in some cases, a sentence has been correctly tagged even if the sentence is hard to comprehend:

- *Some of the consequences of that new standard of international English will be that some or all grammatical changes will be made.*
- *One of the main advantages would be a better communication along everybody.*

These sentences are semantically somewhat confusing, but syntactically sound: simple constructions, well-known words. NLTK did not make a single mistake with either of the sentences. However, manual analysis of these was relatively difficult. Hence, from learner language perspective and those interested in using the tagged corpus to find irregularities in the data, the automatic tagger does not flag semantically nonsensical sentences, as long as the syntax is similar to the native language that the automatic tagger has been trained on.

Even is most often used as an adjective, but in the following example, based on context, it has been correctly tagged as an adverb, probably because it is followed by an adjective and the most frequent premodifier for an adjective is an adverb

- *International firms and services will help to make life easier and maybe* ***even*** *cheaper.*

In most cases, NLTK correctly tags words that have typos, In the following examples, *singel*, loose and *Englis* have been correctly tagged as an adjective, verb and a singular proper noun, respectively:

- *While it would be a good thing, to have a* ***singel*** *language to use at any given time. I think that we as humans would* ***loose*** *too much*
- *Also if this new standard international* ***Englis*** *emerge to other countries, it gives people chance to compare it with the regular English*

However, when presented with a more complicated typo, e.g. a gerund (a verb form which functions as a noun) that should be tagged as an adjective, it is tagged as a verb:

- *Another disadvantage would be the change of economy. Even though we may hope for cheaper prices and equality, the* ***globalasing*** *economy could also raise the cost of everything and lead to capitalism.*

In theory, the verb tag could be considered correct, however, the tagger does not make this debatable mistake in other equivalent cases, e.g. the word *everything* is correctly tagged as a noun in essay 1. *I'm*, in the following sentence, is sliced apart during the data cleaning stage using the code presented earlier in the thesis and then analyzed as *I* and *m*, and tagged as personal pronoun and modal, respectively. NLTK did not handle this situation well, as *am* is not a modal:

- *Also, **I'm** not sure if those countries could study that new standard international English, because they could be fond of their own national language*

However, the exact outcome may be due to how my script slices words, and it is also an inherent problem with this kind of tagging. One of the prevailing questions with automatic tagging is how should several words, when represented as one word, be tagged.

Dialect in this sentence is tagged as a proper noun, which I, understandably, tagged incorrect. However, this common noun is presented without a determiner, making it, effectively from the perspective of the automating tagging a likely candidate for a proper noun. One possible additional explanation for this error is that the word is capitalised.

- ***Dialect** could be hardly understandable and then it is hard for people to socialize*

The cases I have thus far presented have made sense. It is easy to understand why a word was tagged as it was, be it correct or incorrect. There are, however, several cases where a word and its corresponding tag seem illogical. For example, there are several instances where the word *that* has been tagged as a wh-determiner:

- *For example, when the new international English is emerging, other languages **that** are not that strong, will disappear and the country will be left with no native language.*
- *In my opinion the main positive aspect of the international English will be communication **that** will change to a lot more easier*
- *Why change something **that** has been working excellent for hundreds of years?*

This demonstrates a clear mistake by NLTK. *That* is a determiner in these cases, but *that* cannot be a wh-determiner. This is also another mistake where there severity of the mistake is under debate. For the present thesis, these cases have been left in as mistakes, but they can be added to the list of exceptions, resulting in an even lower error percentage.

In the following sentence, *languages* is tagged as a 3rd person singular present verb:

- *I believe that **languages** as they are today should be left exactly the same.*

I was confused at first, but it can be understood as a verb form representing a verb base and 3rd person singular: *Does he language? Oh, he languages all the time*. Another peculiarity is why the word *change* in the following sentence has been tagged as singular or mass noun, while it is clearly a verb:

- Why **change** something that has been working excellent for hundreds of years?

The structure of the sentence points to the fact that the second word should be a verb, as *why* is followed by a verb in most cases.

NLTK mostly tags nouns and adjectives correctly, but some peculiarities arise even when tagging these types of words. For example, *international* is tagged as singular proper noun in the following sentence:

- ***International** firms and services will help to make life easier and maybe even cheaper.*

Furthermore, the word *international* appears numerous times in the essay where that sentence is from and from other essays, yet this is the only occurrence of it with an incorrect tag. One possible explanation for this error is that the word is capitalised.

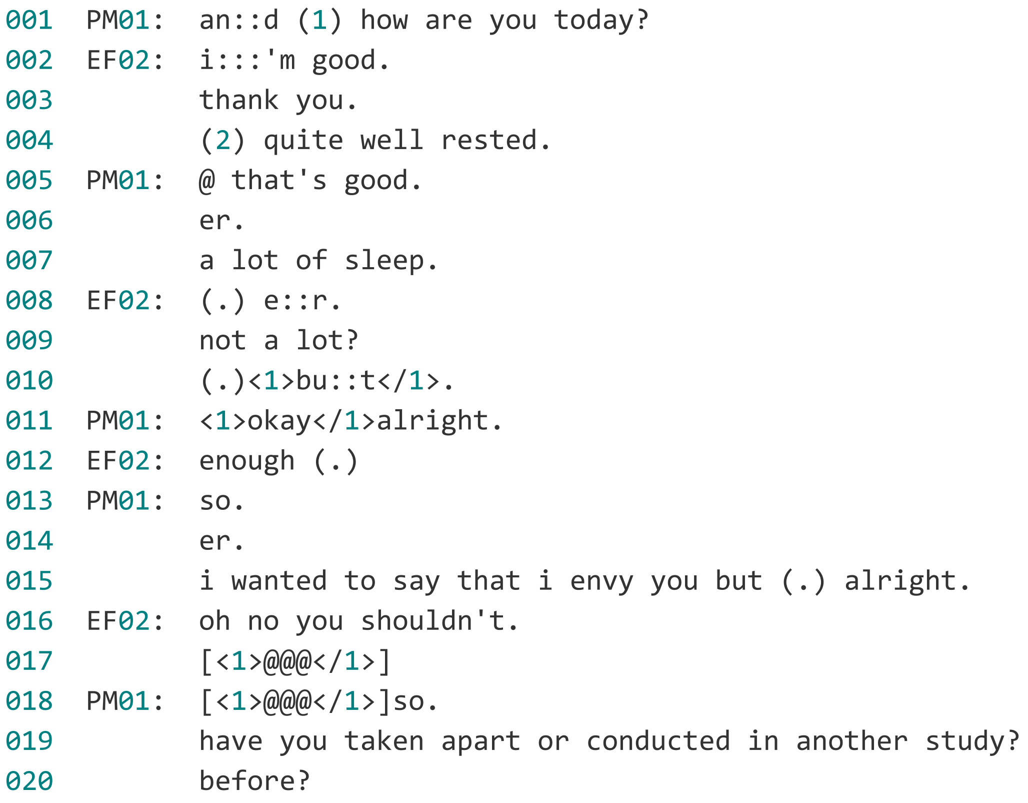
The word *emerge* is tagged by NLTK as a singular noun in the following sentence:

- *Also if this new standard international Englis* ***emerge*** *to other countries, it gives people chance to compare it with the regular English.*

What is even more peculiar about this sentence is that that *Englis* is tagged correctly and there seem to be no other problems when tagging this sentence.

2.4.2. Overview of spoken texts

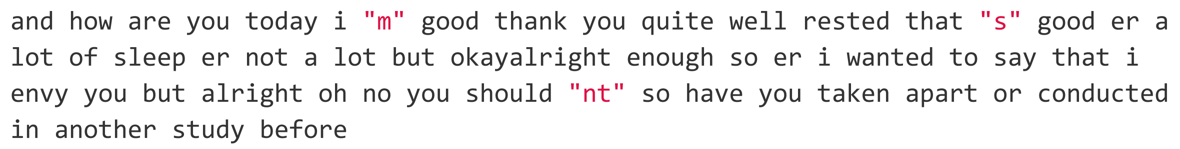
As explained in the section about Automatic tagging of spoken texts, these interviews contain large amounts of “junk data” that needs to be parsed out before NLTK can tag it. Original transcribed spoken texts in the TCELE are formatted as shown in Figure 8.



```
001  PM01:  an::d (1) how are you today?
002  EF02:  i:::'m good.
003         thank you.
004         (2) quite well rested.
005  PM01:  @ that's good.
006         er.
007         a lot of sleep.
008  EF02:  (.) e::r.
009         not a lot?
010         (.)<1>bu::t</1>.
011  PM01:  <1>okay</1>alright.
012  EF02:  enough (.)
013  PM01:  so.
014         er.
015         i wanted to say that i envy you but (.) alright.
016  EF02:  oh no you shouldn't.
017         [<1>@@@</1>]
018  PM01:  [<1>@@@</1>]so.
019         have you taken apart or conducted in another study?
020         before?
```

Figure 8. Example of the original transcribed interview in TCELE

The result of the initial parsing script is given in Figure 9. It was decided to lump the speech produced by both speaker together into one continuous contextualised string of text.



```
and how are you today i "m" good thank you quite well rested that "s" good er a
lot of sleep er not a lot but okayalright enough so er i wanted to say that i
envy you but alright oh no you should "nt" so have you taken apart or conducted
in another study before
```

Figure 9. Output of the parsed spoken text

If the text as a whole makes little semantic sense, the value of automatic tagging decreases. The results are not as informative. Nevertheless, I manually analysed a single sample of 229 automatic tags. Spoken texts were only analyzed with the added exceptions, no previous analysis exists to compare improvements. Results are as follows:

1. NLTK error rate: **28.38**. Word count: **229**

It is important to note that it is possible to improve the initial parsing script to produce more valuable results, such as parsing it line by line, but it was decided that improvements would be minor, since lines themselves are still incomplete sentences and contain numerous filler words. The error rate would decrease, but only slightly. Given that the automatic tagger was fed a lumped together piece of text, it is surprising that it managed to tag with such high accuracy. For the automatic tagging of spoken data one would need to do manual coding of the data and chunk it into clauses. As a way to put the automatic tagging of spoken Estonian learner English into perspective, we can compare it with the error rate of spoken texts in BNC corpora which was 1.17% before and 2.00% after removing ambiguity (BNC manual).

2.4.3. In depth analysis of interviews

If sentences are incomplete, the tagger produces a significant amount of peculiar tags. This section will present some of these anomalies. Out of 229 tagged words, 12 are either instances of variations of “er”, “erh”, or “mhmh”. If we simply remove those 12 tags, we get an error rate of **23.14%** (down from 28.38%). An improvement, but still not low enough to allow for any kind of meaningful use. The first suspicious error can be found on line 15:

- 015 i wanted to say that i envy you but (.) **alright**.

In this case, *alright* is tagged as past tense verb. It is a peculiar mistake, *alright* can never be classified as a verb, especially not a past tense verb. It is always either an adjective or an adverb. It must be noted that NLTK analysed the text as a whole and not by lines, however, the preceding words are still the same. Interestingly, this reveals something about NLTK’s tagging algorithm: it very rarely analyses words as structural units by themselves, and mostly relies on sentence-level analysis.

In the 229 words analysed, *I* (pron; Used to refer to oneself as speaker or writer.) is tagged as a noun on four separate occasions, once as a proper noun. The occasions are:

1. 002 EF02: **i**:::'m good.
2. 021 EF02: (1) y:::::es **i** have.
3. 061 EF02: **i**::::::'m:::: studying to become a te a teacher and **i** decided it on the teacher's day er i::n:: my er=

The third example sentence has *I* tagged as a noun on two separate occasions. Even more, the *m* following the *I* in two cases is tagged correctly as a non-3rd person singular present verb.

In the following example, *something* is tagged as a verb, gerund or present participle:

- 032 EF02: SOMEthing like tha::t yea.

This makes little structural, syntactic or semantic sense. Blame cannot be put on the tagger's preference to analyse based on sentence structure, rather than word class (as explained earlier), as *something* is the first word of the sentence. Also, *something* can never be classified as a verb, *something* simply cannot exist as a verb. The only logical conclusion is that it takes capitalization into account and therein lies the issue.

In conclusion, tagging the spoken text corpora, in its current form, provides no valuable information. Improving the script would decrease the error percentage slightly, but for it to be successfully automatically tagged, it needs to be changed manually, so it would be more coherent.

2.4.4. Types of errors

Based on the types of errors presented in Table 3 in de Haan (2000 : 74), a similar table is created for the types of errors found in the corpora tagged for this thesis. One of the observations made is that that for the present set of learner English, Estonian learners do not make many such mistakes that confuse the tagger.

The table contained the following types of errors:

1. obvious keyboard error
2. word class transfer
3. verb morphology error
4. grammatical error
5. L1-lexis related
6. L1-morphology-related
7. L1-spelling-related
8. L1-pronunciation-related
9. L1-pronunciation-related
10. Hypercorrection

As explained in the in-depth analysis chapter, NLTK does not make mistakes when tagging these kinds of errors (except for a single case, when the word *globalasing*, a verb morphology error, was tagged as a verb). Additionally, this is definitely a contributing factor as to why NLTK achieves such a low error rate. As a further note of caution, it should be kept in mind that exceptions were allowed. Potentially, spoken texts can be predicted to contain very few such errors, as they are usually written down by someone with more knowledge of the language, not a learner.

Another comparison can be made of NLTK against the study by van Rooy and Schäfer (200) where three different taggers were compared: TOSCA, CLAWS and Brill. However, a comparison here is of little value, as the underlying corpora are and sample size are different and the automatic taggers used are different. Still, one crucial observation can be made - differently from the CLAWS tagger, the NLTK tagger does not have problems with spelling errors.

2.4.5. Counting tags

As the types of tags used and their frequency is important when comparing different texts, it was necessary to write a small script that would take the input of all the written essays and spoken texts and output the count of tags. Figure 10 presents the shortened script that automates the counting logic for both written and spoken TCELE corpora (this is not the complete script, unnecessary lines, such as filenames and output print, removed for brevity).

A line that has been automatically tagged and manually analysed is in the following format: *of (IN) - preposition +*. This script extracts the word and the automatic tag, not paying attention to Words API results or the result of the manual analysis. Lines 1-6 are essentially just a way to loop (loops execute a block of code a number of times) over the essays and interviews, and lines in those essays and interviews that have been automatically tagged and manually analysed. Line 7 contains the check whether the line is an actual tag of a word or some junk or a punctuation mark, e.g. a line containing a punctuation mark will be: *.) - +*. As can be seen, that line does not contain both parenthesis, it is therefore filtered out. Lines 8-9 replace other junk data generated by the scripts that analyse the data, e.g. a line can contain punctuation marks: *, which (WDT) - +*. These are all removed. Line 10 removes surrounding whitespaces and splits the line by inner whitespaces. The result is an array (a list) of strings (text elements) of each line.

```
1.  for name in filenames:
2.          with open(name, 'r') as file:
3.                  lines = file.readlines()
4.                  essay_tag_counts = {}
5.
6.                  for line in lines:
7.                          if "(" in line and ")" in line:
8.                                  entities = line.replace(".", "").replace(",",
9.  "").replace(":", "").replace("?", "")
10.                                 entities = entities.strip().split(" ")
11.
12.                                 word = entities[0]
13.                                 tag = entities[1]
14.
15.                                 if tag in essay_tag_counts:
16.                                         essay_tag_counts[tag] += 1
17.                                 else:
18.                                         essay_tag_counts[tag] = 1
```

```
19.
20.                              if tag in total_tag_counts:
21.                                     total_tag_counts[tag] += 1
22.                              else:
23.                                     total_tag_counts[tag] = 1
24.
25.               sorted_counts = sorted(essay_tag_counts.items(),
26. key=operator.itemgetter(1), reverse=True)
27.
28. sorted_counts = sorted(total_tag_counts.items(), key=operator.itemgetter(1),
29. reverse=True)
```

Figure 10. Shortened script (Script 5) for counting the frequency of tags

Lines 12-13 is where the extraction takes place. The first element of the array will always be the word itself and the second element of the array will be NLTK's automatic tag of that word. After the necessary data has been extracted, the word and the tag is, firstly, added to a dictionary (a list consisting of keys and values, rather than just elements) related to a certain essay or interview and, secondly, added to a dictionary of total counts. If a dictionary key contains a tag, the count is incremented, else a new key is added to the dictionary. Finally, the dictionaries are ordered (sorted) according to the count of specified tags.

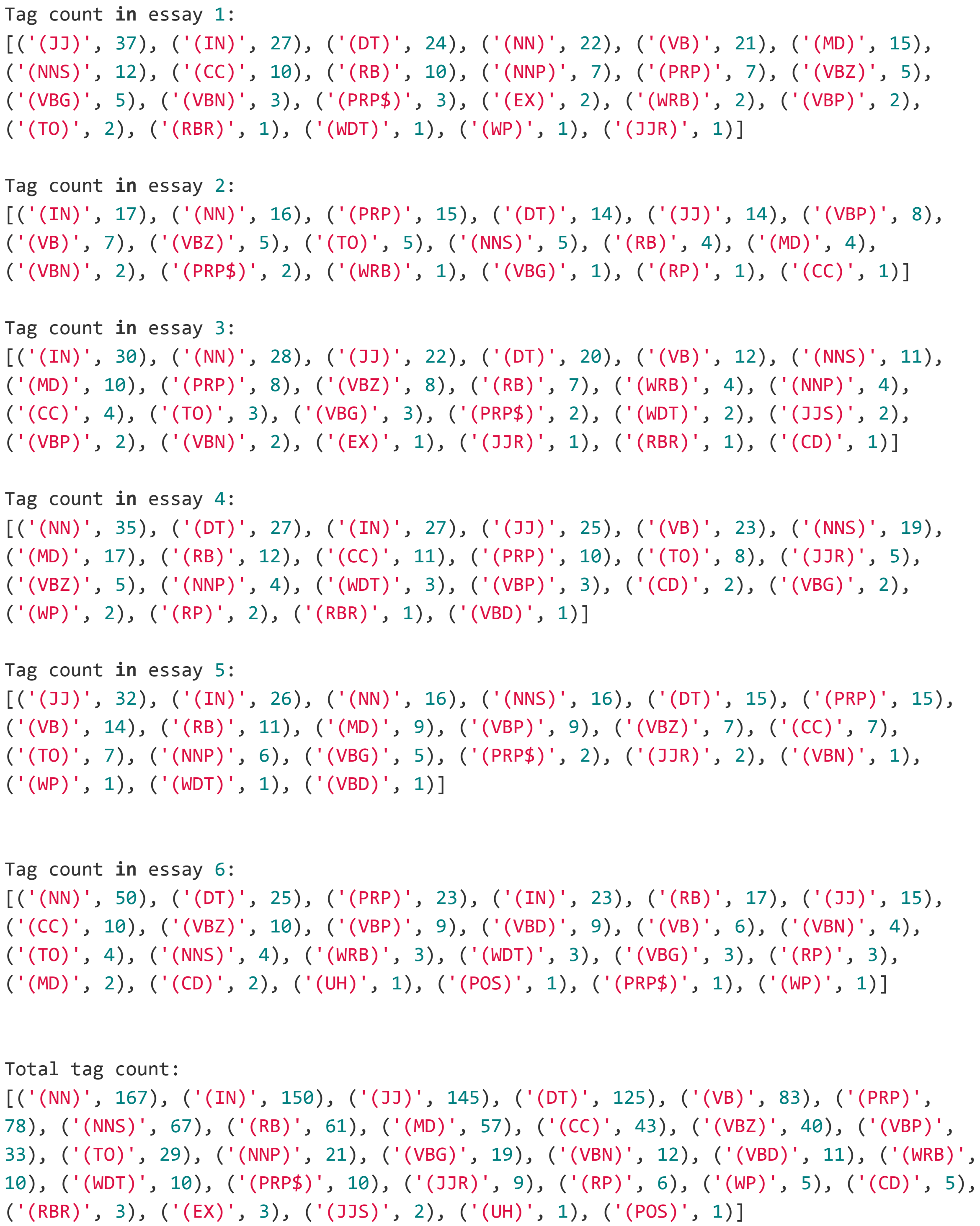
The output for the five essays is given in Figure 11. The output is in the form of lists of counts, separated by parentheses, the initial character is the shortcut for the tag and is followed by, separated by a comma, the occurrence frequency. Essay 6 means in this context the excerpt of the automatically analysed spoken text from the interview.

Penn-Treebank tagset uses “shortcuts” for tags, so this script does not display the full description of the tag. The full list is given in the section titled *Penn-Treebank tagset*. The most popular tags are the following:

1. **JJ** – Adjective
2. **NN** –Noun, singular or mass
3. **IN** – Prep. or sub. conjunction
4. **DT** – Determiner
5. **PRP** – Personal pronoun
6. **RB** – Adverb
7. **VB** – Verb, base form
8. **NNS** – Noun, plural

The results are relatively consistent: the five most common tags of each essay are the aforementioned eight tags. There are three different most prevalent tags: (1) adjective, (2) preposition or subordinating conjunction (3) noun, singular or mass. The most prevalent tag in each of the five essays and the interview is as follows:

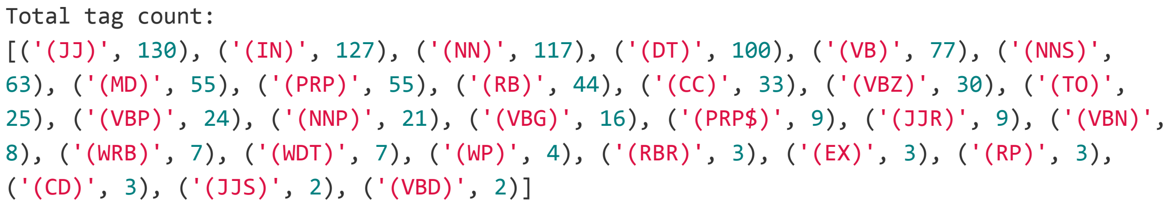
1. **Essay 1**: Adjective
2. **Essay 2**: Preposition or subordinating conjunction
3. **Essay 3**: Preposition or subordinating conjunction
4. **Essay 4**: Noun, singular or mass
5. **Essay 5**: Adjective
6. **Interview**: Noun, singular or mass
7. **Overall**: Noun, singular or mass



```
Tag count in essay 1:
[('(JJ)', 37), ('(IN)', 27), ('(DT)', 24), ('(NN)', 22), ('(VB)', 21), ('(MD)', 15),
('(NNS)', 12), ('(CC)', 10), ('(RB)', 10), ('(NNP)', 7), ('(PRP)', 7), ('(VBZ)', 5),
('(VBG)', 5), ('(VBN)', 3), ('(PRP$)', 3), ('(EX)', 2), ('(WRB)', 2), ('(VBP)', 2),
('(TO)', 2), ('(RBR)', 1), ('(WDT)', 1), ('(WP)', 1), ('(JJR)', 1)]

Tag count in essay 2:
[('(IN)', 17), ('(NN)', 16), ('(PRP)', 15), ('(DT)', 14), ('(JJ)', 14), ('(VBP)', 8),
('(VB)', 7), ('(VBZ)', 5), ('(TO)', 5), ('(NNS)', 5), ('(RB)', 4), ('(MD)', 4),
('(VBN)', 2), ('(PRP$)', 2), ('(WRB)', 1), ('(VBG)', 1), ('(RP)', 1), ('(CC)', 1)]

Tag count in essay 3:
[('(IN)', 30), ('(NN)', 28), ('(JJ)', 22), ('(DT)', 20), ('(VB)', 12), ('(NNS)', 11),
('(MD)', 10), ('(PRP)', 8), ('(VBZ)', 8), ('(RB)', 7), ('(WRB)', 4), ('(NNP)', 4),
('(CC)', 4), ('(TO)', 3), ('(VBG)', 3), ('(PRP$)', 2), ('(WDT)', 2), ('(JJS)', 2),
('(VBP)', 2), ('(VBN)', 2), ('(EX)', 1), ('(JJR)', 1), ('(RBR)', 1), ('(CD)', 1)]

Tag count in essay 4:
[('(NN)', 35), ('(DT)', 27), ('(IN)', 27), ('(JJ)', 25), ('(VB)', 23), ('(NNS)', 19),
('(MD)', 17), ('(RB)', 12), ('(CC)', 11), ('(PRP)', 10), ('(TO)', 8), ('(JJR)', 5),
('(VBZ)', 5), ('(NNP)', 4), ('(WDT)', 3), ('(VBP)', 3), ('(CD)', 2), ('(VBG)', 2),
('(WP)', 2), ('(RP)', 2), ('(RBR)', 1), ('(VBD)', 1)]

Tag count in essay 5:
[('(JJ)', 32), ('(IN)', 26), ('(NN)', 16), ('(NNS)', 16), ('(DT)', 15), ('(PRP)', 15),
('(VB)', 14), ('(RB)', 11), ('(MD)', 9), ('(VBP)', 9), ('(VBZ)', 7), ('(CC)', 7),
('(TO)', 7), ('(NNP)', 6), ('(VBG)', 5), ('(PRP$)', 2), ('(JJR)', 2), ('(VBN)', 1),
('(WP)', 1), ('(WDT)', 1), ('(VBD)', 1)]


Tag count in essay 6:
[('(NN)', 50), ('(DT)', 25), ('(PRP)', 23), ('(IN)', 23), ('(RB)', 17), ('(JJ)', 15),
('(CC)', 10), ('(VBZ)', 10), ('(VBP)', 9), ('(VBD)', 9), ('(VB)', 6), ('(VBN)', 4),
('(TO)', 4), ('(NNS)', 4), ('(WRB)', 3), ('(WDT)', 3), ('(VBG)', 3), ('(RP)', 3),
('(MD)', 2), ('(CD)', 2), ('(UH)', 1), ('(POS)', 1), ('(PRP$)', 1), ('(WP)', 1)]


Total tag count:
[('(NN)', 167), ('(IN)', 150), ('(JJ)', 145), ('(DT)', 125), ('(VB)', 83), ('(PRP)',
78), ('(NNS)', 67), ('(RB)', 61), ('(MD)', 57), ('(CC)', 43), ('(VBZ)', 40), ('(VBP)',
33), ('(TO)', 29), ('(NNP)', 21), ('(VBG)', 19), ('(VBN)', 12), ('(VBD)', 11), ('(WRB)',
10), ('(WDT)', 10), ('(PRP$)', 10), ('(JJR)', 9), ('(RP)', 6), ('(WP)', 5), ('(CD)', 5),
('(RBR)', 3), ('(EX)', 3), ('(JJS)', 2), ('(UH)', 1), ('(POS)', 1)]
```

Figure 11. Output of the frequency of automatic tags in the five essays and the transcribed interview from TCELE

As can be seen, overall, the singular or mass noun is the most popular tag with a staggering 167 occurrences (out of 977, 17% of total tags), however, the interview is a wildcard as its error rate is close to 20%, and it adds 50 occurences of a singular or mass noun. As we saw from the analysis of error, the pronoun *I* was very often tagged as a noun. It would be wise to count tag frequency without including the interview. When removing the interview from the total count, there is a different ranking for the frequency of the tags as seen from Figure 12.



```
Total tag count:
[('(JJ)', 130), ('(IN)', 127), ('(NN)', 117), ('(DT)', 100), ('(VB)', 77), ('(NNS)',
63), ('(MD)', 55), ('(PRP)', 55), ('(RB)', 44), ('(CC)', 33), ('(VBZ)', 30), ('(TO)',
25), ('(VBP)', 24), ('(NNP)', 21), ('(VBG)', 16), ('(PRP$)', 9), ('(JJR)', 9), ('(VBN)',
8), ('(WRB)', 7), ('(WDT)', 7), ('(WP)', 4), ('(RBR)', 3), ('(EX)', 3), ('(RP)', 3),
('(CD)', 3), ('(JJS)', 2), ('(VBD)', 2)]
```

Figure 12. Frequency of automatic tags assigned by NLTK for the five essays sampled from TCELE (*n* = 977)

As can be seen from Figure 12, without the interview, the singular or mass noun is not the most popular tag, it is not even the second most popular one. One would think the most popular tag would either be IN (preposition or subordinating conjunction) or DT (determiner), but JJ (adjective) is the most popular tag among the essays automatically tagged and manually analysed, with IN being a close second and NN (noun, singular or mass) or DT not falling fall behind.

These were just the five essays (and one interview) manually analysed for this thesis. The same logic can be applied to all of the 127 automatically tagged essays. Instead of reading data from the five analysed essays, data is read from the original source file that contains all tagged essays. The script to do this is given in Figure 13.

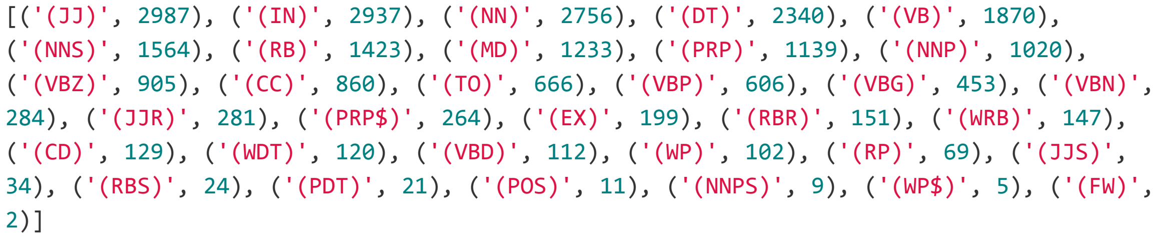
```
1.  for line in lines:
2.          line = line.strip()
3.  
4.          if (line != ""):
5.                  entities = line.replace(".", "").replace(",",
6.  "").replace(":", "").replace("?", "")
7.                  entities = entities.strip().split(" ")
8.  
9.                  tag_counter = 0
10. 
11.                 tagged_word = ""
12. 
13.                 for entity in entities:
14. 
15.                         if (entity == ""):
16.                                 continue
17. 
18.                         if "(" in entity and ")" in entity:
19.                                 tag = entity
20.                                 if tag in tag_counts:
21.                                         tag_counts[tag] += 1
22.                                 else:
23.                                         tag_counts[tag] = 1
24.                         else:
25.                                 word = entity.lower()
26.                                 if word in word_counts:
27.                                         word_counts[word] += 1
28.                                 else:
29.                                         word_counts[word] = 1
```

Figure 13. Script 6 from the analysis

As the text as a whole contains empty lines between essays, it is first necessary to check whether a line is blank, that is done on line 4 of the script. The essays are, from a technical standpoint, on a single line, even though the text editor wraps the essays to multiple lines. If a line is not blank, it is an essay. Next, replace all punctuation marks, as was done earlier. After that, it is necessary to split the essays into words and tags (line 7). Essays are formatted as:

```
The (DT) positive (JJ) is (VBZ) that (IN) it (PRP) /.../
```

It is relatively easy to split the essay by whitespaces, then every other string will be a word, followed by the tag of that word. However, in the current scenario, it is not necessary to associate a certain word with a certain tag. So, if a string contains parenthesis (line 18), it is a tag, then add to the dictionary of tags, otherwise (line 24) add it to the dictionary of words. This produces two dictionaries, one of words, and the other of tags. The dictionaries are then sorted and the output is printed. The results of the tag dictionary are given in Figure 14.



```
[('(JJ)', 2987), ('(IN)', 2937), ('(NN)', 2756), ('(DT)', 2340), ('(VB)', 1870),
('(NNS)', 1564), ('(RB)', 1423), ('(MD)', 1233), ('(PRP)', 1139), ('(NNP)', 1020),
('(VBZ)', 905), ('(CC)', 860), ('(TO)', 666), ('(VBP)', 606), ('(VBG)', 453), ('(VBN)',
284), ('(JJR)', 281), ('(PRP$)', 264), ('(EX)', 199), ('(RBR)', 151), ('(WRB)', 147),
('(CD)', 129), ('(WDT)', 120), ('(VBD)', 112), ('(WP)', 102), ('(RP)', 69), ('(JJS)',
34), ('(RBS)', 24), ('(PDT)', 21), ('(POS)', 11), ('(NNPS)', 9), ('(WP$)', 5), ('(FW)',
2)]
```

Figure 14. Frequency of automatic tags assigned by NLTK for the 127 essays in TCELE (*n* = 24,733)

As can be seen, when counting the tags of all 127 essays, JJ (adjective) is still the most prevalent tag, however, IN (preposition or subordinating conjunction) is a close second and DT (determiner) is the fourth most prevalent tag. The fact that the category of adjective is the most popular tag may be due to the nature of the TCELE texts - these are essays written as an entry exam for the MA programme at the University of Tartu, Department of English Studies and they are based on an original text. However, more detailed analysis is required to verify this suspicion.

The results of the word dictionary are numerous and it is worth pointing out that there are several words with only one occurence. The top 20 words used are as follows (capitalised and lower-case words were included in the counts):

1. the: 1045
2. english: 847
3. of: 671
4. to: 666
5. it: 658
6. and: 626
7. a: 577
8. is: 557
9. be: 492
10. that: 485
11. language: 482
12. in: 408
13. would: 379
14. will: 374
15. new: 326
16. international: 269
17. for: 256
18. languages: 248
19. other: 246
20. people: 243

Prepositions, conjunctions and determiners such as *the, of, to, and, a, in* are prevalent, as expected, however, words like *English, language, international, people, languages* are highly specific to the TCELE corpus. The original text that the essays were based upon dealt with the future of the English language – hence, there are a lot of occurrences of the adjective *English* in the learner texts. This also explains why JJ (adjective) is the most popular tag, as *English* is sometimes an adjective and *international* is always an adjective, and they are both represented in the top twenty tagged words.

The total number of words used in the written text corpora is 24,733 (i.e. the number of tokens), the total number of different words is 2,176 (i.e. the number of types). Penn-Treebank tagset consists of 36 different tags, however, in the essays, only 33 different tags were used. The essays do not contain the following tags:

1. LS (List item marker)
2. SYM (Symbol)
3. UH (Interjection)

2.5. Discussion

The experiment has been an astounding success, the final error rate for written text is below 2%, even lower than other taggers have achieved and what was predicted based on those taggers (cf. Manning 2011). However, the error rate for spoken texts is greater than 20%, thus, automatic tagging cannot be meaningfully applied to the spoken subcorpus of TCELE in the current format. The error rate reported for the written subcorpus of TCELE is similar to what van Rooy and Schäfer (2002) found when they automatically tagged their corrected corpora. "Corrected" meaning that they manually analysed the errors and made corrections to problematic sentences. NLTK achieved such a low error rate (lower than 2%) without any manual correction, except for the few exceptions allowed as described in the thesis. This warrants explanation as to the possible reasons why NLTK performed so well. There are three initial guesses as to why NLTK out-performed other taggers:

1. NLTK uses the Penn-Treebank tagset which contains only 36 tags; thus the present study made use of fewer tags compared to CLAWS's 137 tags or TOSCA's 220 tags which were used in the previous studies.
2. Brill tagger does use Penn-Treebank tagset, but differently from NLTK, Brill is purely rule-based.
3. There were a few exceptions made for this study during the automatic tagging process. The exceptions are as follows:
    a. If a word has been tagged as as a verb, but an incorrect type of verb form (the tense must be correct), the tagger has been correct

b. If superlative or comparative adjectives have been tagged as “just” adjectives, the tagger has been correct
c. If “there” has been tagged as an existential there, and it is a pronoun, the tagger has been correct
d. If Numbers, even when used as adjectives or nouns, are tagged as “cardinal number”, the tagger has been correct

The author of this thesis would like to point out that an error rate of below 2% is probably lower than a manual tagger would achieve on the first reading. Computers do not, yet, know all the nuances of human language, but humans become tired, distracted and make mistakes. It is very probable that if two humans were to tag the same text, their tags would differ, making manual tagging subjective. For example, Manning (2011: 172) discusses the case where two human annotators had an interannotator disagreement rate as high as 7.2%. Computers may be wrong, but they are consistent. Moreover, they do not get tired nor distracted. Some mistakes made by automatic taggers, however, are easily fixable if the tagger’s corpora and algorithms are improved, e.g. the word *alright* was tagged as a verb on the following line of the spoken subcorpus: *i wanted to say that i envy you but (.)* ***alright****.* This reveals that NLTK’s tagging algorithm very rarely analyses words as structural units by themselves, and mostly relies on sentence-level analysis.

This warrants further research, as results could be improved if word-level tagging was done on top of sentence-level tagging. The word-level tagging corpora needs to be extensive, contain all possible part-of-speech tags for words, so it would not make mistakes, as it would only overwrite the tags of words where the automatic tagger has been absolutely wrong, e.g. *alright* as a verb. This would, of course, only marginally decrease the error rate. For example, the word *languages* was once tagged as a verb by NLTK, and when writing in a very informal or colloquial style (almost sarcastic) *languages* can be used as a verb: *Does he language? Oh, he languages all the time*.

However, this corpus (at least the written subcorpus) does not contain such colloquial language and, while there is an error rate, it is insignificant and does not warrant manual analysis for all the essays. There are two ways to continue with the project:

1. Manual analysis of all the essays
2. Create an interactive corpus as is, with an admin interface so errors could be found and fixed continuously, when found

As mentioned, manual analysis will take a significant amount of time and still does not guarantee that tags will be error-free, as humans are prone to make mistakes. The second option is preferential, as language is complicated by nature and a 0% error rate on initial tagging is difficult to achieve, either way. However, the error rate decreases significantly when we make a few exceptions to tagging rules or train the tagger. It would decrease even more if fewer tags are used. The essential tagset should be decided on before automatic tagging takes place and should depend on the needs of the potential users of the tagged corpus .

One of the issues that merits discussion, is the fluctuation of the error rate across the five essays. The error rates in the five manually analysed essays is as follows:

- Essay 1: NLTK error rate: **0.45%**. Word count: 221
- Essay 2: NLTK error rate: **0.00%**. Word count: 123
- Essay 3: NLTK error rate: **2.65%**. Word count: 189
- Essay 4: NLTK error rate: **1.22%**. Word count: 245
- Essay 5: NLTK error rate: **0.98%**. Word count: 204

Such fluctuations, from 0% in essay 2 to 2.65% in essay 3 can be, mostly, explained by the tagging of *that* as a wh-determiner discussed in the analysis section. Namely, essay 3 has three different instances of *that* being tagged as a wh-determiner and one instance in essay 1. If *that* as a wh-determiner was to be added to the list of exceptions, the rate would be as follows:

- Essay 1: NLTK error rate: **0.0%**. Word count: 221
- Essay 2: NLTK error rate: **0.0%**. Word count: 123
- Essay 3: NLTK error rate: **1.59%**. Word count: 189
- Essay 4: NLTK error rate: **1.22%**. Word count: 245
- Essay 5: NLTK error rate: **0.98%**. Word count: 204

The combined error rate would fall to **0.758%**. The author of this thesis believes the error rate would fall even further when manual analysis were done on all the essays, as most errors seem to be completely random. The following is the complete list of NLTK errors from essays 3 and 4, where the error rate is above 1%:

1. Essay 3:

    a. **so** as an **adverb** (possible, but incorrect in context)
    b. **languages** as as a **verb** (nonsensical)
    c. **change** as a **noun** (possible, but incorrect in context)

2. Essay 4:

    a. **International** as a **proper noun** (nonsensical)
    b. **globalasing** as a **verb** (nonsensical)
    c. **good** as a **verb** (nonsensical)

As one can see, most of these are nonsensical. It can be expected that such nonsensical exceptions do not occur in every essay. Otherwise, this is a good sign, as nonsensical tags are easier to fix by adding rules and checks for word types. *so* as an adverb and *change* as a noun are more complicated errors, as they are realistic tags.

Overall, Words API was not as useful a tool as initially anticipated. While the free tier does allow to tag the entire corpora over the course of a few days, the tag set of Words API is nowhere near as capacious as Penn-Treebank tagset. The service does not list its tag set on the website and they did not reply to emails when the question was asked, but its tag set seems to consist only of the most basic tags (noun, verb, adjective, adverb, determiner). Additionally, Words API queries find results only if the request contains the base form of the word (or, if it is another form, then the result can be junk). Finally, Words API offers tags out of context, so if a word has numerous possible tags, all the tags are returned and the result is relatively useless. Words API is still a great service, and not irrelevant in the current context, but not as useful as initially anticipated.

“How many different tags should be used when tagging this corpora, so the outcome could be most sufficient?” is a question that should have been asked before tagging takes place. Is the purpose of this corpora to have the lowest error rate, or should it have a more complex tag set? These are crucial questions and there is no right or wrong answer - a balance needs to be struck and depends on the needs of the potential users of the tagged corpus. In the case of TCELE, the expected users are researchers interested in Estonian learner English, teachers of English and student of English in Estonia. Therefore, for research purposes the tagset should be fairly detailed, but for teachers and students, it should be fairly simple.

As mentioned in the Taggers section, in the BNC corpora, removing ambiguities (meaning possible multiple-interpretations, making sentences “prettier”) actually increased the error rate. TCELE corpora features many such ambiguities, and the low error rate may be thanks, or due, to these ambiguities. Unfortunately, analysis of ambiguities falls out of the scope of this thesis, however, this is an area of possible further research.

NLTK had problems with contractions, e.g. *I’m* was separated into two words and and tagged as a personal pronoun and a modal, respectively. Setting aside the fact that NLTK has problems with contractions, the question remains: how should multiple words, represented as one, be tagged? Phrase tokens are a possible solution, but then phrase tokens should be used everywhere, not only in such cases, and that would change the outcome of the analysis as a whole. The problem is also described in the BNC manual:

> There are, however, exceptions to this. A single orthographic word may contain more than one grammatical word: e.g. in the case of verb contractions and negative contractions such as she's, they'll, we're, don't, isn't, two tags are assigned in sequence to the same orthographic word. Also quite frequent is the opposite circumstance, where two or more orthographic words are given a single grammatical tag: e.g. compound conjunctions such as so that and as well as are each assigned a single conjunction tag, and likewise compound prepositions such as instead of and up to are each assigned a single preposition tag. Naturally, whether such orthographic sequences should be treated as single word for grammatical tagging purposes depends on the context. As well as in some contexts is not a conjunction, but a sequence of adverb - adverb - conjunction/preposition. Up to in some contexts is a sequence of adverbial particle - preposition.
>
> (BNC Manual http:// www.natcorp.ox.ac.uk/docs/gramtag.html)

Since the experiment has been a success, achieving a staggering **1.06%** combined error rate, it makes sense to publish the tagged written text corpora to a closed database where it can be interactively accessed via an API. In computer programming, an application programming interface, or API, is a set of subroutine definitions, protocols, and tools for building application software. In layman's terms, it is just a set of instructions for accessing the exact data you require, e.g. *www.corpora.com/api/get_essays* would be the api to get a list of essays, whereas *www.corpora.com/api/get_words?id=123* would be the api to access the words and tags of an essay with the id 123.

The database cannot be accessible to the general public, as it contains potentially sensitive information, therefore a security system needs to be implemented as well. The easiest way to achieve this is to add a required api key to every request. The server only returns the data if the api key is correct. In more complicated software projects, each user receives their own api key, so the server knows exactly who accesses what, however, in the case of viewing this corpora data, a single api key is secure enough.

In order to publish the TCELE subcorpus of written essays discussed in this thesis, it must first be formatted correctly. Before it was parsed and formatted with the purpose of simplifying manual analysis, now it must be simplified for the server to understand and correctly save the information to a database. Also, to identify essays, it is important to extract the unique identifier and bind it to the essays (or, technically, the words of an essay).

First, the data is read from the word document, as was done when parsing it for analysis. Then, instead of tagging the words and saving them to a file, the json (a lightweight data-interchange format) of the tagged words is stored in memory. An essay's unique identifier is also stored in that json. When all the data is formatted into that json, it is sent to the server to be stored in a database. The server extracts the data from the json format and maps it to database tables, rows and columns, accordingly. To simplify this process, an ORM (object relational mapping) tool called RedBean was used. To be able to search for a tag, a word or an essay by its identifier, "essay" as a concept, as a collection of words, does not exist. This written word corpus database consists of 27,404 tagged word entries (this is not the word count, punctuation marks are also entries) and 127 essay entries. A word entry is in the format specified in Figure 15.

```
{
   "id":"1",
   "essay_id":"1148001",
   "tag":"DT",
   "text":"Some",
   "index":"0"
}
```

Figure 15. Word entry in the database of the tagged TCELE corpus

In addition to the essay_id, tag and text, it also has an index, which is essentially the order number of a word, so the words could later be formed into a coherent text to the end user. It also has an id, which is the unique identifier used for storing it in an SQL database. An essay entry is in the format given in Figure 16.

```
{
   "id":"1",
   "identifier":"01148001"
}
```

Figure 16. Essay entry in the database of the tagged TCELE corpus

It must only contain its unique database identifier and unique essay identifier. The “id” field is for internal storage use only, it has no value to the user. If one were to search for an essay by its unique identifier, the query (in pseudocode, simplified SQL) would look like something as the following:

```
SELECT EVERYTHING FROM WORDS WHERE ESSAY_ID = 123
```

“WORD” in this is the table (a table is a set of data elements using a model of vertical columns and horizontal rows) and the user selects all the rows from that table where the essay identifier is “123”. If a user were to query for a list of essays, the query would look something like the following:

```
SELECT EVERYTHING FROM ESSAY
```

Notice that there is no “where” case. Every essay identifier is returned.

As mentioned earlier, storing each word separately is important when the user wants to make a more complicated search. For example, if one wants to select all the determiners of a given essay, the query would be:

```
SELECT EVERYTHING FROM WORDS WHERE TAG = 'NOUN' AND ESSAY_ID = 123
```

Or, say, if the user wants to select all the instances of the word “and” from all essays:

```
SELECT EVERYTHING FROM WORDS WHERE TEXT = 'and'
```

When the data is stored and the api is documented, any developer can use this database to create their own client (website or mobile application) that makes queries against the database. This is part is not just theoretical, this section has been put into practice. The data is stored in a server and available via the API. The two urls are:

- https://corpus.nikitech.eu/data/list.php?api_key=<api_key>
- https://corpus.nikitech.eu/data/details.php?api_key=<api_key>&&id=<id>

Replace the <api_key> with a valid API key and in the second url, also the <id> with a valid id, e.g.:

- https://corpus.nikitech.eu/data/list.php?api_key=f34869jdfg
- https://corpus.nikitech.eu/data/details.php?api_key=f34869jdfg&&id=01148001

There is also a client available at: https://corpus.nikitech.eu/. Enter an API key at the top right corner and press “Submit”, a list of essays should appear. An API for search does not yet exist. Please note that the web client is experimental, it was developed on Google Chrome browser on macOS, support for other operation systems and browsers has not been verified. Email aare.undo@gmail.com to request for a valid API key.

All of the scripts used for the empirical study of the thesis are available in GitHub (a web-based hosting service for version control using git; mostly used for code.) under the account of the author: https://github.com/Nikituh/scripts/tree/master/nltk

Overall, the author of this thesis is under the impression that the sample size (five essays, part of one interview) is not large enough to get a reliable result. It would require at least 10% (13 essays, chosen at random) of the corpus to be analysed, and the same essays analysed by multiple people to ensure everything is correctly tagged, in order to correctly determine the average error rate for the entire written corpus. The spoken interview needs to be re-formatted in order to be automatically tagged.

Finally, this thesis aimed to find out which of the following scenarios is correct:

1. The success rate of the automated part-of-speech tagger is higher for native language, because there are errors/ innovations in learner language.
2. The success rate of the automated part-of-speech tagger for learner language is on a par with native language.
3. The success rate of the automated part-of-speech tagger is higher for learner language, because the learners' language is structurally less complex.

In the context of TCELE, the most probable conclusion is that the correct answer is the third scenario: automated part-of-speech tagging is simplified by the fact that learners' language is structurally less complex. However, as another area of further research, this needs to be more thoroughly investigated with other similar learner language corpora.

Conclusion

The aim of this thesis was to review previous research, provide the benefits and drawbacks of automatic tagging, then automatically tag (using automatic part-of-speech tagger software) and manually analyse (manually review the automatically tagged texts) TCELE (Tartu Corpus of Estonian Learner English) written and spoken text subcorpora, and, finally, analyse the results of automatic tagging, find and explain the calculated error rate. Questionable areas, such as the appropriate amount of tags that should be used, how to deal with contractions, and problems with automatic taggers were also discussed, in greater or lesser detail.

The software used to tag the TCELE corpus was NLTK (Natural Language Toolkit), an open-source library (a “library” in computer science is, essentially, a piece of code written and distributed by someone else, to be used by others) written in the Python programming language. NLTK features an extensive internal corpora 3.5GB in size and sophisticated tagging algorithms. NLTK was compared to TOSCA-ICLE tagger, CLAWS tagger and Brill tagger, summarising the research by van Rooy and Schäfer (2002), but since the corpora in question were different, objective comparison is difficult. For a comprehensive comparison all of the different taggers should be used with one and the same dataset. Still, some general observations can be made. NLTK achieved results similar to CLAWS (approximately 2% error rate), whereas the error rates of TOSCA-ICLE and Brill were higher. Van Rooy and Schäfer (2002) randomly selected five essays of roughly 400 words from the entire existing corpus, for a sample of just more than 2,000 words, or 1.25% of the current TLE (Treebank of

Learner English) corpus, whereas NLTK analysed ~1,000 words, or roughly 4% of TCELE corpora.

As a result of the empirical study carried out in this thesis, 127 written essays (~200-300 words each, ~25,000 words in total) and one spoken interview (~2000 words) were automatically tagged. For the manual analysis and evaluation of the automatic tagging, 5 written essays and 229 words of the spoken interview were randomly selected. After careful analysis, it was discovered that the initial error rate for the three first sampled essays was quite high: 14.4%, 7.3% and 8.9%, respectively. The following four exceptions were added:

1. If a word has been tagged as as a verb, but an incorrect type of verb (the tense must be correct), the tagger has been correct
2. If superlative or comparative adjectives have been tagged as "just" adjectives, the tagger has been correct
3. If "there" has been tagged as an existential there, and it is a pronoun, the tagger has been correct
4. If Numbers, even when used as adjectives or nouns, are tagged as "cardinal number", the tagger has been correct

After accounting for these new rules, the adjusted error rates for the first three essays were 0.45%, 0.00% and 2.65% and the error rate for the additional two essays was 1.22% and 0.98%. As for spoken texts, the initial error rate was 28.38%, however, out of 229 tagged words, twelve are either instances of variations of "er", "erh", or "mhmh". If we simply remove those twelve tags, the error rate falls down to 23.14%. While a 5% fall is impressive, the adjusted error rate of 23.14% is still nowhere near acceptable. The error rate could be (marginally) improved by improving the parsing script, but the language is still too erratic, further analysis of spoken texts was abandoned at this point.

The frequency of tags and words was also counted. Over the five essays manually analysed, the most prevalent tag of each was as follows: Essay 1: Adjective; Essay 2: Preposition or subordinating conjunction; Essay 3: Preposition or subordinating conjunction; Essay 4: Noun, singular or mass; Essay 5: Adjective.

Over all 127 written texts, the five most prevalent tags were as follows: Adjective: (130), preposition or subordinating conjunction (127), noun, singular or mass (117), determiner (100), verb, base form (77). However, the adjective being the most prevalent tag is highly specific to this corpora. The theme of the essay plays as a major role in the frequency of tags.

The 20 most popular words used in the essays are as follows: the (1045), english (847), of (671), to (666), it (658), and (626), a (577), is (557), be (492), that (485), language (482), in (408), would (379), will (374), new (326), international (269), for (256), languages (248), other (246), people (243).

Finally, opportunities for further research and the compilation of an interactive corpus interface were provided. The tagged corpora were uploaded to a server hosted by the author of this thesis. The data is available at https://corpus.nikitech.eu/. Email aare.undo@gmail.com to receive an API Key to view the tagged corpora.

References

BNC2 POS-tagging Manual. http://www.natcorp.ox.ac.uk/docs/bnc2error.htm, accessed 14 May 2018.

Daniel, Anna. 2015. *The Use of Adjectives and Adverbs in Estonian and British Student Writing: A Corpus Comparison*. Master's thesis. University of Tartu. Available at http://hdl.handle.net/10062/47055.

De Haan, Pieter. 2000. *Tagging non-native English with the TOSCA–ICLE tagger*. In Mair, Ch. & M. Hundt (eds.) *Corpus linguistics and linguistic theory*. 69–79.

Crystal, David. 2003. *Rediscover Grammar*. Harlow: Longman

Documentation, NLTK. Available at: https://www.nltk.org/, accessed 21 January 2018.

Documentation, OpenNLP. Available at: https://opennlp.apache.org/docs/, accessed 30 March 2018.

Garside, Roger, Geoffrey Leech & Anthony McEnery (eds.). 1997. *Corpus annotation: Linguistic information from computer text corpora*. Harlow: Longman.

Granger, Sylviane. 2012. *Learner Corpora*. In *The Encyclopedia of Applied Linguistics*. Oxford: Blackwell.

Kirsimäe, Merli. 2017. *The Compilation and Lexicogrammatical Analysis of an Estonian Spoken Mini-Corpus of English as a Lingua Franca*. Master's thesis. University of Tartu. Available at http://hdl.handle.net/10062/57562.

Kennedy, Graeme D. 1998. *An Introduction to Corpus Linguistics.* Harlow: Longman.

Leech, Geoffrey. *A Brief Users' Guide to the Grammatical Tagging of the British National Corpus*. http://www.natcorp.ox.ac.uk/docs/gramtag.html, accessed 14 May 2018

Leech, Geoffrey. 2005. *Adding Linguistic Annotation*, in M. Wynne, *Developing Linguistic Corpora: a Guide to Good Practice* (Oxford: Oxbrow Books), pp. 17-29.

Manning, Christopher D. Manning. 2011. *Part-of-speech tagging from 97% to 100%: is it time for some linguistics?*. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 171-189). Springer, Berlin, Heidelberg.

Manning, Christopher D & Schuetze, Hinrich. 2001. *Maximum Entropy Modeling*. In *Foundations of Statistical Natural Language Processing*. MIT Press. Cambridge, Massachusetts; London, England.

Merilaine, Elina. 2015. *The frequency and variability of conjunctive adjuncts in the Estonian–English Interlanguage Corpus*. Master's thesis. University of Tartu. Available at http://hdl.handle.net/10062/47065.

Müller, Christoph & Michael Strube. 2006. *Multi-Level Annotation of Linguistic Data with MMAX2*. In: Sabine Braun, Kurt Kohn, Joybrato Mukherjee (eds.), *Corpus technology and language pedagogy. New Resources, new tools, new methods,* 197-214. Frankfurt am Main: Peter Lang.

Nesselhauf, Nadja. 2004. *Learner corpora and their potential for language teaching*. In *How to use corpora in language teaching?* 12, 125-156.

Rayson, Paul Edward. 2015. *Computational tools and methods for corpus compilation and analysis*. In Douglas Biber and Randi Reppen (eds.), *The*

Cambridge Handbook of English corpus linguistics, 32-49. Cambridge: Cambridge University Press.

Schmid, Helmut. 1994. *Probabilistic Part-of-Speech Tagging Using Decision Trees*. In *Proceedings of International Conference on New Methods in Language Processing*. Manchester, UK. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf., accessed 29 September 2017.

Stanford Natural Language Processing Group, The. *Stanford Log-linear Part-Of-Speech Tagger*. Available at: https://nlp.stanford.edu/software/tagger. shtml, accessed 20 February 2018.

Sutherland, Joseph L. *doc2text 0.2.4*. Available at: https://pypi.python.org/pypi/doc2text, accessed 15 April 2018.

Tammiste, Lenne. 2016. *The use of adjective-noun, verb-noun and phrasal-verb-noun collocations in Estonian learner corpus of English*. Master's thesis. University of Tartu. Available at http://hdl.handle.net/10062/53280.

University Centre for Computer Corpus Research on Language, BNC Page. Available at: http://ucrel.lancs.ac.uk/bnc2/bnc2error.htm, accessed 30 April 2018.

University Centre for Computer Corpus Research on Language, CLAWS Page. Available at: http://ucrel.lancs.ac.uk/claws/, accessed 30 April 2018.

Van Den Heuvel, Theo. 1998. *Literary and Linguistic Computing*, Volume 3, Issue 3, pp 147–151.

Van Halteren, Hans. 1999. *Performance of taggers*. In H. Van Halteren (Ed.), *Syntactic Wordclass Tagging*, pp. 81-94. Dordrecht: Springer.

Van Rooy, Bertus, & Schäfer, Lande. 2002. *The effect of learner errors on POS tag errors during automatic POS tagging*. In *Southern African Linguistics and Applied Language Studies*, *20*(4), 325-335.

Van Rooy, Bertus. 2015. *Annotating learner corpora*. In S. Granger, G. Gilquin, & F. Meunier (Eds.), *The Cambridge Handbook of Learner Corpus Research* (Cambridge Handbooks in Language and Linguistics, pp. 79-106). Cambridge: Cambridge University Press.

Appendix 1: Automatically tagged sample from TCELE

Written essay 1

Some (DT) of (IN) the (DT) consequences (NNS) of (IN) that (DT) new (JJ) standard (NN) of (IN) international (JJ) English (NNP) will (MD) be (VB) that (IN) some (DT) or (CC) all (DT) grammatical (JJ) changes (NNS) will (MD) be (VB) made (VBN) . Also (RB) , the (DT) languages (NNS) of (IN) other (JJ) countries (NNS) might (MD) turn (VB) into (IN) English-like (JJ) mixtures (NNS) . Other (JJ) positive (JJ) sides (NNS) of (IN) that (DT) would (MD) be (VB) that (IN) people (NNS) might (MD) communicate (VB) much (RB) more (RBR) easily (RB) . But (CC) , there (EX) can (MD) be (VB) various (JJ) negative (JJ) sides (NNS) in (IN) that (DT) new (JJ) standard (NN) . For (IN) example (NN) , when (WRB) the (DT) new (JJ) international (JJ) English (NNP) is (VBZ) emerging (VBG) , other (JJ) languages (NNS) that (WDT) are (VBP) not (RB) that (IN) strong (JJ) , will (MD) disappear (VB) and (CC) the (DT) country (NN) will (MD) be (VB) left (VBN) with (IN) no (DT) native (JJ) language (NN) . Also (RB) , the (DT) original (JJ) \u2018 (NNP) native (JJ) \u2019 (NNP) english (NN) will (MD) disappear (VB) too (RB) , leaving (VBG) that (IN) easy (JJ) , international (JJ) new (JJ) English (NNP) to (TO) spread (VB) and (CC) to (TO) destroy (VB) everything (NN) what (WP) is (VBZ) left (VBN) from (IN) the (DT) old (JJ) English (NNP) . My (PRP$) opinion (NN) is (VBZ) that (IN) it (PRP) can (MD) be (VB) both (DT) positive (JJ) and (CC) negative (JJ) for (IN) the (DT) world (NN) and (CC) the (DT) language (NN) itself (PRP) . It (PRP) might (MD) be (VB) damaging (VBG) for (IN) other (JJ) languages (NNS) but (CC) it (PRP) also (RB) might (MD) be (VB) a (DT) good (JJ) start (NN) for (IN) better (JJR) communication (NN) for (IN) different (JJ) countries (NNS) all (DT) over (IN) the (DT) world (NN) . My (PRP$) side (NN) in (IN) that (DT) situation (NN) is (VBZ) neutral (JJ) . I (PRP) \u2019 (VBP) d (RB) be (VB) glad (JJ) , when (WRB) other (JJ) countries (NNS) would (MD) get (VB) along (IN) finally (RB) but (CC) in (IN) the (DT) same (JJ) time (NN) , it (PRP) makes (VBZ) me (PRP) sad (JJ) , because (IN) there (EX) would (MD) be (VB) no (DT) native (JJ) language (NN) anymore (RB) . But (CC) in (IN) the (DT) meantime (NN) , let (VB) \u2019 (NNP) s (VB) be (VB) proud (JJ) for (IN) having (VBG) our (PRP$) own (JJ) language (NN) and (CC) being (VBG) different (JJ) . Noone (NN) can (MD) predict (VB) the (DT) future (NN) of (IN) new (JJ) international (JJ) language (NN) .

Written essay 2

The (DT) positive (JJ) is (VBZ) that (IN) it (PRP) brings (VBZ) mankind (NN) under (IN) a (DT) singel (JJ) language (NN) . It (PRP) removes (VBZ) the (DT) languge (NN) barrier (NN) , that (IN) we (PRP) have (VBP) , when (WRB) a (DT) person (NN) wishes (VBZ) to (TO) work (VB) in (IN) a (DT) foreign (JJ) country (NN) . Unfortunatly (RB) by (IN) creating (VBG) a (DT) singel (JJ) language (NN) that (IN) we (PRP) all (DT) use (VBP) , we (PRP) would (MD) destroy (VB) many (JJ) in (IN) the (DT) process (NN) . Also (RB) many (JJ) cultures (NNS) would (MD) be (VB) destroyed (VBN) . Due (JJ) to (TO) the (DT) fact (NN) that (IN) culture (NN) is (VBZ) a (DT) huge (JJ) part (NN) of (IN) any (DT) culture (NN) . While (IN) it (PRP) would (MD) be (VB) a (DT) good (JJ) thing (NN) , to (TO) have (VB) a (DT) singel (JJ) language (NN) to (TO) use (VB) at (IN) any (DT) given (VBN) time (NN) . I (PRP) think (VBP) that (IN) we (PRP) as (IN) humans (NNS) would (MD) loose (VB) too (RB) much (JJ) . Even (RB) if (IN) we (PRP) don (VBP) \u2019 (JJ) t (NNS) say (VBP) it (PRP) out (RP) loud (JJ) and (CC) at (IN) times (NNS) we (PRP) say (VBP) that (IN) we (PRP) hate (VBP) our (PRP$) language (NN) . In (IN) our (PRP$) heaths (NNS) we (PRP) hold (VBP) it (PRP) dear (JJ) to (TO) us (PRP) .

Written essay 3

When (WRB) it (PRP) is (VBZ) likely (JJ) that (IN) a (DT) new (JJ) standard (NN) of (IN) international (JJ) English (NNP) will (MD) emerge (VB) , there (EX) might (MD) be (VB) some (DT) of (IN) the (DT) consequences (NNS) for (IN) English (NNP) and (CC) also (RB) for (IN) other (JJ) languages (NNS) . In (IN) my (PRP$) opinion (NN) the (DT) main (JJ) positive (JJ) aspect (NN) of (IN) the (DT) international (JJ) English (NNP) will (MD) be (VB) communication (NN) that (WDT) will (MD) change (VB) to (TO) a (DT) lot (NN) more (JJR) easier (JJ) . In (IN) that (DT) case (NN) the (DT) understanding (NN) between (IN) different (JJ) nations (NNS) will (MD) be (VB) great (JJ) and (CC) people (NNS) would (MD) be (VB) much (RB) more (RBR) enthusiastic (JJ) about (IN) learning (VBG) something (NN) new (JJ) . Every (DT) one (CD) of (IN) us (PRP) is (VBZ) interested (JJ) in (IN) new (JJ) relationships (NNS) from (IN) abroad (RB) so (RB) why (WRB) not (RB) to (TO) learn (VB) something (NN) that (IN) everyone (NN) knows (VBZ) ? The (DT) negative (JJ) aspect (NN) from (IN) this (DT) situation (NN) is (VBZ) a (DT) big (JJ) , but (CC) the (DT) only (JJ) one (NN) : fading (NN) other (JJ) languages (NNS) . If (IN) most (JJS) of (IN) the (DT) people (NNS) would (MD) be (VB) able (JJ) to (TO) communicate (VB) with (IN) other (JJ) nations (NNS) of

(IN) the (DT) world (NN) then (RB) why (WRB) should (MD) they (PRP) talk (VB) with (IN) someone (NN) in (IN) their (PRP$) mother-language (NN) ? For (IN) most (JJS) of (IN) the (DT) people (NNS) it (PRP) seems (VBZ) like (IN) a (DT) waste (NN) of (IN) time (NN) . I (PRP) believe (VBP) that (DT) languages (VBZ) as (IN) they (PRP) are (VBP) today (NN) should (MD) be (VB) left (VBN) exactly (RB) the (DT) same (JJ) . Everything (NN) is (VBZ) working (VBG) great (JJ) with (IN) today (NN) \u2019 (NNP) s (NN) lifestyle (NN) and (CC) we (PRP) should (MD) keep (VB) it (PRP) in (IN) that (DT) way (NN) . Why (WRB) change (NN) something (NN) that (WDT) has (VBZ) been (VBN) working (VBG) excellent (NN) for (IN) hundreds (NNS) of (IN) years (NNS) ?

Written essay 4

The (DT) new (JJ) standard (NN) of (IN) international (JJ) English (NNP) will (MD) have (VB) a (DT) lot (NN) consequences (NNS) , which (WDT) will (MD) affect (VB) all (DT) countries (NNS) in (IN) the (DT) long (JJ) perspective (NN) . One (CD) of (IN) the (DT) main (JJ) advantages (NNS) would (MD) be (VB) a (DT) better (JJR) communication (NN) along (IN) everybody (NN) . International (NNP) firms (NNS) and (CC) services (NNS) will (MD) help (VB) to (TO) make (VB) life (NN) easier (JJR) and (CC) maybe (RB) even (RB) cheaper (JJR) . Also (RB) , the (DT) invention (NN) of (IN) useful (JJ) machinary (NN) would (MD) grow (VB) because (IN) of (IN) the (DT) teamwork (NN) . Another (DT) advantage (NN) of (IN) the (DT) new (JJ) standard (NN) would (MD) be (VB) better (JJR) and (CC) new (JJ) knowledge (NN) of (IN) different (JJ) cultures (NNS) . This (DT) would (MD) make (VB) people (NNS) more (RBR) tolerant (JJ) towards (NNS) each (DT) other (JJ) and (CC) also (RB) it (PRP) may (MD) decrease (VB) violence (NN) and (CC) wars (NNS) would (MD) be (VB) rare (JJ) events (NNS) . Although (IN) it (PRP) might (MD) seem (VB) to (TO) be (VB) a (DT) welcoming (JJ) standard (NN) , it (PRP) also (RB) has (VBZ) some (DT) negative (JJ) consequences (NNS) with (IN) it (PRP) . One (CD) of (IN) the (DT) disadvantages (NNS) would (MD) be (VB) the (DT) growth (NN) of (IN) globalisation (NN) . It (PRP) is (VBZ) a (DT) danger (NN) to (TO) small (JJ) cultures (NNS) and (CC) different (JJ) traditions (NNS) . Another (DT) disadvantage (NN) would (MD) be (VB) the (DT) change (NN) of (IN) economy (NN) . Even (RB) though (IN) we (PRP) may (MD) hope (VB) for (IN) cheaper (JJR) prices (NNS) and (CC) equality (NN) , the (DT) globalasing (VBG) economy (NN) could (MD) also (RB) raise (VB) the (DT) cost (NN) of (IN) everything (NN) and (CC) lead (NN) to (TO) capitalism (NN) . In (IN) conclusion (NN) , the (DT) new (JJ) standard (NN) of (IN) international (JJ) English (NNP) seems (VBZ) quite (RB) frightening (JJ) , but (CC) also (RB) a (DT) new (JJ) way (NN) to (TO) new (JJ) solutions (NNS) , which (WDT) may (MD) please

(VB) the (DT) people (NNS) . Because (IN) of (IN) the (DT) English (NNP) spreading (NN) changes (NNS) are (VBP) coming (VBG) and (CC) those (DT) , who (WP) refuse (VBP) to (TO) keep (VB) up (RP) with (IN) the (DT) \u201c (JJ) trend (NN) \u201d (NN) will (MD) soon (RB) find (VB) themselves (PRP) in (IN) difficulties (NNS) and (CC) also (RB) be (VB) bitter (JJ) about (IN) everything (NN) . It (PRP) is (VBZ) important (JJ) to (TO) keep (VB) up (RP) with (IN) everything (NN) , which (WDT) at (IN) first (JJ) place (NN) seems (VBZ) to (TO) be (VB) strange (JJ) , because (IN) we (PRP) never (RB) know (VBP) , what (WP) good (VBD) it (PRP) might (MD) bring (VB) .

Written essay 5

In (IN) my (PRP$) opinion (NN) the (DT) negative (JJ) effects (NNS) of (IN) this (DT) new (JJ) standard (JJ) international (JJ) English (NNP) could (MD) be (VB) origin (VBN) of (IN) new (JJ) words (NNS) . Within (IN) the (DT) new (JJ) words (NNS) the (DT) also (RB) can (MD) be (VB) changes (NNS) in (IN) grammar (NN) , because (IN) every (DT) language (NN) is (VBZ) different (JJ) and (CC) all (DT) of (IN) them (PRP) consist (VBP) different (JJ) kind (NN) of (IN) difficulties (NNS) . Also (RB) I (PRP) \u2019 (VBP) m (MD) not (RB) sure (JJ) if (IN) those (DT) countries (NNS) could (MD) study (VB) that (IN) new (JJ) standard (JJ) international (JJ) English (NN) , because (IN) they (PRP) could (MD) be (VB) fond (NN) of (IN) their (PRP$) own (JJ) national (JJ) language (NN) . But (CC) I (PRP) hope (VBP) that (DT) is (VBZ) not (RB) a (DT) various (JJ) threat (NN) . On (IN) a (DT) positive (JJ) side (NN) I (PRP) think (VBP) that (IN) is (VBZ) good (JJ) if (IN) more (JJR) and (CC) more (JJR) people (NNS) get (VBP) to (TO) know (VB) English (JJ) language (NN) . Knowing (VBG) many (JJ) different (JJ) languages (NNS) only (RB) helps (VBZ) people (NNS) while (IN) they (PRP) are (VBP) travelling (VBG) around (IN) the (DT) world (NN) , helping (VBG) tourist (NN) by (IN) giving (VBG) them (PRP) informations (NNS) or (CC) directions (NNS) . Also (RB) if (IN) this (DT) new (JJ) standard (JJ) international (JJ) Englis (NNP) emerge (NN) to (TO) other (JJ) countries (NNS) , it (PRP) gives (VBZ) people (NNS) chance (NN) to (TO) compare (VB) it (PRP) with (IN) the (DT) regular (JJ) English (NNP) . Then (RB) you (PRP) can (MD) decide (VB) whether (IN) you (PRP) want (VBP) to (TO) learn (VB) it (PRP) or (CC) not (RB) . Also (RB) what (WP) I (PRP) think (VBP) could (MD) be (VB) negative (JJ) consequence (NN) is (VBZ) that (IN) with (IN) the (DT) new (JJ) international (JJ) English (NNP) , which (WDT) emerge (VBP) over (IN) the (DT) world (NN) , people (NNS) could (MD) be (VB) starting (VBG) to (TO) use (VB) new (JJ) accents (NNS) and (CC) it (PRP) \u2019 (NNP) s (VBD) tough (JJ) to (TO) deal (VB) with (IN) . Dialect (NNP) could (MD) be

(VB) hardly (RB) understandable (JJ) and (CC) then (RB) it (PRP) is (VBZ) hard (JJ) for (IN) people (NNS) to (TO) socialize (VB) .

Spoken interview 1

and (CC) how (WRB) are (VBP) you (PRP) today (NN) i (NN) "m" (VBP) good (JJ) thank (NN) you (PRP) quite (RB) well (RB) rested (VBN) that (DT) "s" (VBZ) good (JJ) er (NN) a (DT) lot (NN) of (IN) sleep (NN) er (NN) not (RB) a (DT) lot (NN) but (CC) okayalright (NN) enough (RB) so (RB) er (NN) i (RB) wanted (VBD) to (TO) say (VB) that (IN) i (JJ) envy (VBP) you (PRP) but (CC) alright (VBD) oh (UH) no (DT) you (PRP) should (MD) "nt" (RB) so (RB) have (VB) you (PRP) taken (VBN) apart (RB) or (CC) conducted (VBN) in (IN) another (DT) study (NN) before (IN) yes (RB) i (NNS) have (VBP) mhmh (NN) what (WDT) was (VBD) that (IN) by (IN) the (DT) way (NN) ehm (NN) it (PRP) was (VBD) a (DT) doctorate (NN) study (NN) of (IN) estonia (JJ) plus (CC) minus (NN) yes (NNS) okay (NN) maybe (RB) sort (NN) of (IN) SOMEthing (VBG) like (IN) that (DT) yea (NN) alright (NN) it (PRP) "s" (VBZ) the (DT) last (JJ) year (NN) for (IN) you (PRP) by (IN) the (DT) way (NN) it (PRP) is (VBZ) it (PRP) is (VBZ) so (RB) and (CC) it (PRP) is (VBZ) right (JJ) before (IN) the (DT) exams (NN) right (NN) yes (NNS) it (PRP) is (VBZ) the (DT) exams (JJ) start (NN) in (IN) a (DT) month (NN) or (CC) so (RB) er (NN) are (VBP) you (PRP) stressed (VBN) out (RP) ehm (NN) stressed (VBD) out (RP) yes (RB) i (JJ) am (VBP) er (RB) do (VBP) you (PRP) like (IN) to (TO) be (VB) in (IN) a (DT) stress (NN) environment (NN) a (DT) little (JJ) a (DT) little (JJ) so (RB) that (DT) "s" (VBZ) that (WDT) was (VBD) the (DT) warm (JJ) up (RP) part (NN) now (RB) the (DT) main (JJ) questions (NNS) er (NN) when (WRB) and (CC) how (WRB) did (VBD) you (PRP) decide (VB) you (PRP) would (MD) want (VB) to (TO) major (JJ) in (IN) whatever (WDT) you (PRP) are (VBP) studying (VBG) at (IN) the (DT) moment (NN) i (NN) "m" (VBP) studying (VBG) to (TO) become (VB) a (DT) te (NN) a (DT) teacher (NN) and (CC) i (NN) decided (VBD) it (PRP) on (IN) the (DT) teacher (NN) "s" (POS) day (NN) er (NN) in (IN) my (PRP$) er (NN) so (IN) it (PRP) "s" (VBZ) a (DT) special (JJ) date (NN) it (PRP) "s" (VBZ) fun (NN) it (PRP) it (PRP) IS (VBZ) a (DT) speacial (JJ) date (NN) what (WP) was (VBD) it (PRP) like (IN) fifth (NN) of (IN) october (NN) mhmh (NN) er (NN) in (IN) two (CD) thousand (NN) and (CC) eight (CD)

RESÜMEE

TARTU ÜLIKOOL
ANGLISTIKA OSAKOND

Aare Undo

Calculating the Error Percentage of an Automated Part-of-Speech Tagger when Analyzing Estonian Learner English – An Empirical Analysis / Automaatse sõnaliigi märgendaja veaprotsendi arvutamine eesti keelt emakeelena kõnelevate inglise keele õppijate korpuse baasil

Magistritöö
2018
Lehekülgede arv: 91

Teksti sõnaliikideks jaotamine sündis koos lingvistikaga, kuid selle protsessi automatiseerimine on muutunud võimalikuks alles viimastel kümnenditel ning seda tänu arvutite võimsuse kasvule. Tekstitöötluse algoritmid on alates sellest ajast iga aastaga üha paranenud. Selle magistritöö raames pannakse üks selle valdkonna lipulaevadest proovile korpuse peal, mis hõlmab eesti keelt emakeelena kõnelevate inglise keele õppijate tekste (TCELE korpus). Korpuse suurus on antud hetkel ca. 25 000 sõna (127 kirjalikku esseed) ning 11 transkribeeritud intervjuud (~100 minutit). Eesmärk on hinnata TCELE ja muude sarnaste korpuste veaprotsenti.

Töö esimeses osas tutvustatakse lugejale korpuse kokkupanemist, annoteerimist ja väljavõtet (ingl. *retrieval*) ning antakse ülevaade sõnaliikide määramisest ja veaprotsendist. Pärast seda antakse ülevaade varasematest uuringutest ning vastatakse muuhulgas, järgnevatele küsimustele: mida on eelnevalt tehtud? Mis olid uuringute leiud? Millised automaatsed märgendajad (ingl. *taggers*) ja sõnaliikide loendeid (ingl. *tagset*) kasutati?

Empiiriline osa tegeleb TCELE tekstide automaatse märgendamise ja automaatse märgendamise käsitsi kontrollimisega. Nii kirjalik kui suuline korpus märgendati automaatselt, kasutades püütoni teeki NLTK (Natural Language ToolKit). Viis esseed vaadati käsitsi üle, et leida automaatse märgendaja veaprotsent. Analüüsi tulemusena leiti, et keskmine kirjaliku teksti veaprotsent on 1.06. Suulise teksti veaprotsent on märkimisväärselt kõrgem – 23.14. Töö viimases osas analüüsiti vigu ja toodi esile problemaatlised kohad. Lisaks võrreldi vigu ja veaprotsenti teiste automaatsete märgjendajatega teistest õppijakeele korpustest. Töö raames toodi välja ka võimalikud variandid edasisteks uuringuteks ning kuidas kasutada antud annoteeritud korpust, sh. algeline variant interaktiivsest korpuseliidesest. Peamine töö panus on see, et leitud veaprotsent on piisavalt madal, et uurijad võivad usaldada TCELE automaatset sõnaliikide märgendust kirjalike tekstide raames. Suulise keele jaoks on vaja põhjalikumat käsitööd. TCELE kirjalik korpus on täielikult sõnaliikide osas märgendatud ja uurijatele kättesaadav.

Märksõnad: sõnaliikide automaatne märgendamine, NLTK, õppijakeele korpus, eesti keelt emakeelena kõnelevate õppijate ingise keel, korpuslingvistika

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Aare Undo,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   Calculating the Error Percentage of an Automated Part-of-Speech Tagger when Analyzing Estonian Learner English – An Empirical Analysis,
   mille juhendaja on Jane Klavan,
   a. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   b. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 15.05.2017

Aare Undo