Universität Tübingen

Master's Thesis in Computational Linguistics

Automatic Prediction of CEFR Proficiency Levels Based on Linguistic Features of Learner Language

Julia Hancke
julia_hancke@yahoo.com

Supervisor: Prof. Dr. Detmar Meurers
Second Examiner: Dr. Frank Richter

29.04.2013

Abstract

In this theses we present an approach to language proficiency assessment for German. We address language proficiency assessment as a classification problem. Our main focus lies on examining a wide range of features on the syntactic, lexical and morphological level. We combine features from previous work on proficiency assessment with proficiency measures from language acquisition research, and additional indicators from readability assessment. Our data set consists of 1027 German short essays that were produced during German as a second language examinations and rated on the CEFR scale by humans examiners. We experiment with different machine learning algorithms and techniques for model optimization, and conduct a number of experiments that mainly investigate the performance of different feature combinations.

Contents

1 Introduction

Tools for the automatic evaluation of free text essays, such as the Educational Testing Service's *e-Rater* (Attali and Burstein, 2006) or Pearson's *Intelligent Essay Assessor* (Landauer et al., 2003), have been an active field of research over the last decade. They are now increasingly employed in high stakes examinations. The most striking advantages of automatic assessment over human raters are speed, lower costs and the reduction of marking inconsistencies.

Automatic language proficiency assessment is a specific case of automatic essay rating that has recently begun to attract attention within the natural language processing community. Briscoe et al. (2010) and Yannakoudakis et al. (2011) automatically graded English as a second or foreign language examinations (Cambridge First Certificate of English). Vajjala and Loo (2013) automatically assigned CEFR[1] proficiency levels to texts written by learners of Estonian.

Language proficiency assessment and automatic essay grading in general mainly differ on the marking criteria they employ. Automatic essay rating systems mainly assign grades based on the content of an essay and some additional stylistic criteria. Many systems have to be specifically trained for each separate essay prompt. In contrast, language proficiency assessment treats an essay mostly as a sample of a student's language skills. The rating an essay receives depends mainly on the language mastery it demonstrates. The marking criteria in previous approaches to automatic proficiency assessment mainly focused on the linguistic properties of learner texts.

In this theses we present an approach to automatic proficiency assessment for German. We address language proficiency assessment as a supervised classification task. We mainly focus on examining a wide range of linguistic features that seem promising indicators of language proficiency. We combined features from previous work on proficiency assessment with insights from research on second language acquisition. Additionally we integrated features from readability assessment and simplification. Those fields are connected to proficiency assessment as they study the comprehension side of texts while proficiency assessment studies the production side.

The empirical basis for our work consists of 1027 German short essays from German as a foreign language exams that were collected by the *MERLIN*[2] project (Wisniewski et al., 2011). All essays were graded by human raters on the scale of the Common European Framework of Reference for

[1] Common European Framework of Reference for Languages

[2] Multilingual Platform for the European Reference Levels: interlanguage Exploration in Context. German is only one of several languages included in the project.

Languages (CEFR), which is gradually becoming a European standard in language assessment. *CEFR level* can stand for the level of an exam that a student takes or for the "grade" an essay receives. In this work we will refer to the first meaning as *exam type* and the second meaning as either *essay rating level* or *proficiency level*.

We conducted several machine learning experiments with our feature set. Initially, we developed a five class classifier with all CEFR *proficiency levels* (A1-C1) as classes. Experiments on five class classification included comparing different algorithms for classification and investigating the predictive power of different feature combinations. Additionally we developed two class "pass of fail" classifiers for each *exam type* separately and used cost sensitive learning for optimizing the results.

In the following section we provide background information on related work and machine learning, and define the linguistic units that occur throughout this theses. Section 3 introduces our data set in more detail. Section 4 describes our preprocessing procedure. In section 5 we discuss the linguistic properties that we used as features for the machine learning experiments, which are then reported on in section 6. Finally section 7 and 8 summarize our results and suggest directions for future work.

2 Background

2.1 Related Work

Although there is - to the best of our knowledge - no previous work on **proficiency assessment** for German, there is research on the automatic grading of English and Estonian language examinations. Briscoe et al. (2010) and Yannakoudakis et al. (2011) used machine learning for automatically grading English as a second language (ESOL) exam scripts. Their experiments were conducted on a set of scripts from the Cambridge Learner Corpus (CLC), a text collection that contains transcripts of short essays produced during First Certificate in English (FCE) exams. The scripts were written in response to a free text prompt.

Briscoe et al. (2010) and Yannakoudakis et al. (2011) deployed several features that mirror linguistic skill such as lexical unigrams and bigrams, part of speech unigrams, bigrams and trigrams, parse rule names, script length and a corpus derived error rate. Yannakoudakis et al. (2011) added grammatical relation (GR) distance measures to this feature set. In both approaches, the RASP system Briscoe et al. (2006) was used for linguistic markup. For determining scores based on their feature set, Briscoe et al. (2010) experimented

with several supervised machine learning techniques such as Support Vector Machine, Time Aggregated Perceptron and Discriminative Preference Ranking. Yannakoudakis et al. (2011) compared Rank Preference Models with Regression Models.

In addition to the features that capture linguistic properties of the text, Briscoe et al. (2010) included Incremental Semantic Analysis (ISA). ISA is a non prompt-specific approximation to content analysis and was added primarily to make the system less vulnerable to score manipulations through superficial text properties like script length. Yannakoudakis et al. (2011) tested their system's vulnerability to subversion by manipulating exam scripts themselves: they swapped words that have the same part-of-speech within a sentence and randomly reordered word unigrams, bigrams and trigrams within a sentence. Additionally, they scrambled the order of sentences within a script. The modified scripts were then assessed by a human examiner and the computer system, and the correlations between the scores were calculated. Except for modifications that included the reordering of entire sentences, the correlations were still good. However the system generally tended to administer higher scores to modified scripts than the human examiner. Yannakoudakis et al. (2011) pointed out that very creative 'outlier' essays, that use stylistic devices, might be rated worse by the system than by a human examiner.

Vajjala and Loo (2013) predicted the proficiency of learners of Estonian based on morpho-syntactic and lexical indicators. Their feature set comprised nominal and adjectival case information as well as verbal tense, mood, voice, number and person. Additionally part-of-speech ratios, several lexical variation measures from SLA, and text length were included. Sequential Minimal Optimization Algorithm was used for classification. Vajjala and Loo (2013) conducted experiments with various feature combinations, feature selection and more sophisticated machine learning techniques like ensemble classifiers and multi-stage cascading classifiers.

For the sake of completeness it should be mentioned that automatic assessment has not only been attempted for written but also for spoken language. **Automatic language proficiency assessment for spoken language** is quite different from the assessment of written productions in a number of ways. Speech recognizers have to be used to process the spoken data. Also, the grading criteria are different. Scores are mostly based on qualities such as fluency, intelligibility and overall pronunciation quality (de Wet et al., 2007).

Bernstein et al. (2000) tried to place learners' spoken responses to prompts on a functional communication scale that is compatible with the Council of

Europe Framework. Content scores were calculated based on correct words. Manner scores were based on pronunciation and fluency. de Wet et al. (2007) assessed fluency based on the rate of speech. While there is hardly any overlap in the features and techniques used in written versus oral proficiency assessment yet, it is thinkable that in the future, criteria from written assessment could be incorporated into spoken language assessment.

In contrast to marking SLA exams, where the marking criteria put a strong emphasis on the linguistic quality of the scripts (Briscoe et al., 2010), **Automated Essay Scoring (AES)** in general is concerned with marking essays on the basis of their style, grammar, organization and often on their relevance to the given task. It is an umbrella term for software that scores written prose (Dikli, 2006). There are several, mostly commercial systems for AES, some of which are deployed to assess high-stakes examination scripts.

One of the earliest system was *Project Essay Grade* (PEG) ((Page, 2003), c.f. Dikli (2006)). It deployed a number of mostly shallow features that approximate properties of well written essays such as fluency, diction, grammar and punctuation (Dikli, 2006). Based on these features PEG used linear regression to first train a model based on pre-examined essays. This model was used to score unrated essays (Yannakoudakis et al., 2011). PEG was criticized for not taking into account the content and organization of the essays. The absence of such features made it easier to trick the system, since superficial features like word and script length can easily be manipulated (Yannakoudakis et al., 2011).

E-Rater (Attali and Burstein, 2006) examines an essay at different levels such as grammar, style, topic similarity and organization. Each text is represented as a weighted vector of those features (Dikli, 2006). Scores are assigned based on the cosine similarity between the vectors of marked training scripts and unmarked scripts (Dikli, 2006). E-Rater also measures relevance to the prompt. This enables the tool to take into account the content of an essay and makes it less vulnerable to score manipulation. The disadvantage is that E-Rater needs to be trained specifically for each prompt.

Intelligent Essay Assessor (IEA) (Landauer et al., 2003) focuses on content analysis based on Latent Semantic Analysis (LSA) (Landauer and Foltz, 1998). Additionally it takes into account grammar and style (Dikli, 2006). The systems needs to be trained on prompt specific texts, and scores are assigned on the basis of similarity between training essays an the essays that should be rated (Dikli, 2006).

The Bayesian Essay Test Scoring System (BETSY) (Rudner and Liang, 2002) approaches AES as a text classification problem. BETSY uses features representing content and style such as the number of words, sentences, verbs,

and commas, and certain content word frequencies to train Bernoulli Naive Bayes models (Dikli, 2006; Yannakoudakis et al., 2011).

More comprehensive surveys of AES systems are offered by Williamson (2009) and Dikli (2006).

As a **tool that measures text complexity**, *Coh Metrix* (Graesser et al., 2004) is relevant for this work although it is neither an AES system, nor a tool for grading second language productions. As indicators of text complexity it deploys about 200 measures of cohesion, linguistic complexity, and readability with an emphasis on cognitively grounded features. The tool outputs the scores for different components of complexity such as lexis, syntax and coherence. However it does not attempt to assign a holistic score on a grading scale. In principle *Coh-Metrix* can be applied for examining the readability of a text as well as for examining a learner text.

Second Language Acquisition (SLA) Studies examine how humans learn languages that are not their native language. Particularly interesting for this work are SLA studies that are linked to language testing or the characterization of proficiency levels. The *Second Language Acquisition & Testing in Europe (SLATE)* network connects a number of such projects that are quite similar to *MERLIN* in purpose. To name just one example, the *English Profile* (`http://www.slate.eu.org/projects.htm#proj2`) aims at offering a more detailed reference for each CEFR level for English. This includes descriptions of the learners' capabilities at different linguistic levels. The project combines corpus linguistics, pedagogy and assessment (Cambridge ESOL)[3].

Additionally this work was informed by several studies that take a computational approach to the examination of language development. Lu (2010) and Biber et al. (2011) automatically extracted and examined specific syntactic constructions as measures of L2 development. Lu (2012) investigated lexical richness measures such as Type-Token Ratio and Lexical Density as indicators of the lexical development of language learners. Crossley et al. (2011b) closely analyzed several lexical scores used in *Coh-Metrix* and measured how well they correlated with human assessment of lexical proficiency.

Readability assessment is relevant to this work because it tries to detect a text's reading level based on its linguistic and sometimes also conceptional and cognitive complexity. Therefore, features that are good indicators for reading level might also be good features for language proficiency as-

[3]Briscoe et al. (2010)'s study was conducted at Cambridge ESOL. The connection is not mentioned explicitly though.

sessment. Recent approaches to readability assessment were mostly bases on statistical natural language processing techniques. Unigram models were deployed for readability classification by Si and Callan (2001) and Collins-Thompson and Callan (2004). Heilman et al. (2007, 2008) added grammatical features to this approach. Schwarm and Ostendorf (2005), Petersen and Ostendorf (2009) and Feng (2010) trained models for readability classification on the *Weekly Reader*, an educational newspaper consisting of articles at four reading levels. They combined traditional features like sentence length and word length with syntactic parse tree features and ngram language models. Feng (2010) additionally deployed discourse features. Vajjala and Meurers (2012) combined features from previous work on readability assessment with lexical diversity measures and parse tree based syntactic measures from research on second language learning.

While the work mentioned above has been conducted on English, there are also approaches that examine readability assessment for other languages. Dell'Orletta et al. (2011) used a mixture of traditional, morpho-syntactic, lexical and syntactic features for building a two class readability model for Italian. Francois and Fairon (2012) built a French readability classifier deploying verb tense and mood along with several other features. *DeLite* readability checker (Vor der Brück and Hartrumpf, 2007; Vor der Brück et al., 2008) is a tool for assessing the readability of German texts that makes use of syntactic, lexical, semantic, discourse and morphological features. However, it is based on a relatively small human annotated corpus of 500 texts that are all at a relatively high reading level. Hancke et al. (2012) combined syntactic, lexical and language model features from previous work on readability assessment on English texts with German specific morphological features. Their two-class readability classifier was trained and tested on a corpus collected from publicly accessible websites for children and adults.

2.2 Machine Learning

Tom Mitchell (Mitchell, 1997) defined machine learning as follows:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

We approach automatic proficiency level assessment as a supervised learning task. Labeled data is used to train models, from which the correct classes of unlabeled data can be predicted. More specifically, as our class labels are on a discrete scale, we use classification. There are five classes corresponding to the CEFR levels A1-C1. If one transformed the labels A1-C1 into the numbers 1-5 one could also apply regression. However we think classification

is better suited as 1-5 are still discrete values.

Throughout this work, *WEKA* machine learning toolkit (Hall et al., 2009) was used. It offers implementations of various machine learning algorithms.

2.2.1 Algorithms for Machine Learning

This section briefly introduces the machine learning algorithms that were used in this work. **Naive Bayes** classifier is a probabilistic learner. It operates on the simplifying assumption that all features are equally important and independent of each other. Predictions are based on Bayes rule of conditional probability :

$$P(H_i|E_j) = \frac{P(E_j|H_i)P(H_i)}{P(E_j)} \tag{1}$$

The conditional probabilities of a piece of evidence (E) given a certain hypothesis (H) are multiplied with each other and with the likelihood of the hypothesis (prior probability). This product is divided by the prior probability of the evidence. The outcome indicates the likelihood of the hypothesis (H) given the evidence (E) (Witten and Frank, 2005). Naive Bayes is easy to interpret and is reported to perform well on many data sets. However it is sensitive to redundancy in the data set (Witten and Frank, 2005).

Decision Trees are a recursive, top-down, divide and conquer approach to classification. The data set is split into subsets by selecting one feature as the root node, making one branch for every possible value of the feature. To decide which node to split on, the amount of information (in terms of entropy) is used, that would be needed to decide on the class of the instances at this node. The information of a node is influenced by the number of instances that reach the node and by the homogeneity of their classes (Witten and Frank, 2005). While decision trees work most naturally with nominal features (Witten and Frank, 2005), they have been extended to also work with numeric data. There are several different brands of decision trees. In this work we use J48, which is the *WEKA* implementation of the popular C.45 decision tree algorithm developed by Ross Quinlan (Witten and Frank, 2005).

Sequential Minimal Optimization Algorithm (SMO) is an effective algorithm for training a Support Vector Machine (Witten and Frank, 2005). Support Vector Machines are an extension of linear models. To find the optimal decision boundary, they fit a hyperplane into the feature space, such that it separates the data points belonging to different classes with the largest possible margin. As hyperplanes can only linearly separate the data, Support Vector Machines map the feature vectors into higher dimensional space using

kernel functions[4]. This mathematical trick enables them to handle non-linear problems with linear models (Witten and Frank, 2005). Support Vector Machines have many advantages. Different kernel functions provide a high degree of flexibility. Support Vector machines are not very sensitive to overfitting, since the large margin hyperplane provides stability (Witten and Frank, 2005).

2.2.2 Evaluation

For a faithful estimate of a machine learner's performance, models should be tested on previously unseen data (Witten and Frank, 2005). It is common practice to split the data into a separate training and test set (**holdout estimation**). An alternative is **cross validation**. The training data is randomly shuffled and equally split into n folds. In **stratified cross validation** the data is sampled such that the class distribution is retained (Witten and Frank, 2005). Successively, one part is left aside while the model is trained on the other $n - 1$ parts and then evaluated on the part that has been left out. The results are averaged over all n iterations (Baayen, 2008). It is common to set $n = 10$, because it has been empirically shown to be a plausible choice for most data sets and learning methods (Witten and Frank, 2005). Cross validation has the advantage, that it uses the data more economically since no test set has to be split off. The fact that it tests the performance on several different partitions, makes it more representative than a single test set. We will use stratified cross validation throughout, and whenever cross validation is mentioned, stratified cross validation is meant.

In cases where there is very little data, **leave-one-out** testing can be useful. This is a special case of cross validation where n equals the total number of instances in the training set. It is a deterministic procedure since no sampling is involved and it allows the most efficient use of small data sets (Witten and Frank, 2005). However it is computationally expensive and stratification is not possible: the correct proportion of classes in the test set cannot be maintained.

Classification results are presented in **accuracy** and **F-measure**. Accuracy reports how many percent of the samples were classified correctly. F-measure represents the trade-off between precision and recall. Precision captures the portion of samples that were correctly classified as a target class. Recall measures the total portion of samples that were classified as a

[4]Kernel functions are basically similarity functions. They are used to efficiently handle the mapping of the feature space into higher dimensions. For basic information on kernels used with SMO see Witten and Frank (2005), for a detailed discussion of kernels in general see Schölkopf and Smola (2003)

target class (Manning and Schütze, 1999).

2.3 Defining Linguistic Units

Most features used for proficiency classification in this work will be based on linguistic units. *Word class* definitions follow the *Stuttgart Tübingen Tagset*. Frequently several tags were combined into more general classes for the implementation. A comprehensive listing can be seen in Table 1. A point worth considering is the definition of 'content word', which we will also refer to as 'lexical token'. In German there is no clear consensus on whether modals are auxiliaries and should be counted as lexical vs. functional units (see Reis (2001) for a discussion). In this work modals are considered to be 'content words' following Reis (2001).

Unit	Definition
lexical token/'content word' (incl. modals)	ADJA, ADJD, ADV, FM, XY, NN, NE, VVFIN, VVIMP, VVINF, VVIZU, VVPP, VMFIN, VMINF, VMPP
lexical token/ 'content word' (excl. modals)	ADJA, ADJD, ADV, FM, XY, NN, NE, VVFIN, VVIMP, VVINF, VVIZU, VVPP
verb	VVFIN, VVIMP, VVINF, VVIZU, VVPP, VAFIN, VAIMP, VAINF, VAPP, VMFIN, VMINF, VMPP
lexical verb (incl. modals)	VVFIN, VVIMP, VVINF, VVIZU, VVPP, VMFIN, VMINF, VMPP
lexical verb (excl. modals)	VVFIN, VVIMP, VVINF, VVIZU, VVPP
finite verb	VVFIN, VVIMP, VAFIN, VAFIN, VMFIN
noun	NN, NE
adjective	ADJA, ADJD
adverb	ADV
conjunction	KOUI, KOUS, KON
wh-Word	PWS, PWAT, PWAV
preposition	APPR, APPRART, APPO, APZR
modifier	ADJA, ADJD, ADV

Table 1: Unit definitions on the level of lexical units according to the *Stuttgart Tübingen Tagset* tags.

Sentence is a common concept, but in fact it is very hard to find a scientific definition that describes it adequately. As Grewendorf et al. (1989) point out, there is no consensus among linguists. Definitions of *sentence* often include

several linguistic levels. Gallmann and Sitta (2004) define *sentence* on three levels: orthographic/phonetic, syntactic and semantic/pragmatic:

> Sätze sind sprachliche Einheiten, die durch die folgenden Merkmale bestimmt sind: [Sentences are linguistic units that are defined by the following characteristics:]
>
> 1. Sie weisen einen bestimmten grammatischen Bau auf. [They have a specific grammatical structure.]
> 2. Sie sind durch die Stimmführung (in der geschriebenen Sprache: durch die Satzzeichen) als abgeschlossen gekennzeichnet. [They are marked as self-contained by prosody (in the written language: by punctuation)]
> 3. Sie sind inhaltliche - relativ - abgeschlossen. [They are - relatively-self-contained regarding their meaning]
> (Gallmann and Sitta, 2004)

For the implementation a more concrete definition is needed: a *sentence* is a unit that is delimited by one of the following sentence final punctuation marks: *.!?*

Unit	Definition
sentence	ROOT
clause	S + 'satzwertige Infnitive'
coordinated clause	CS

dependent clause w. subord. conj.	`(@S`
	`[ <1 KOUS|KOUI|PRELAT|PRELS|PRELS-SB`
	`|PRELS-OA|PRELS-DA|PWAT|PWS|PWAV]`
	`| [< (@NP < PRELAT|PRELS|PRELS-SB|`
	`PRELS-OA|PRELS-DA|PWAT|PWS|PWAV)])`
	`>>  S|CS|DL & !> CS`

relative clause	`(@S`
	`[ <1 PRELAT|PRELS|PRELS-SB|PRELS-OA|PRELS-DA]`
	`| [< (@NP < PRELAT|PRELS|PRELS-SB|PRELS-OA`
	`|PRELS-DA|PWAT|PWS|PWAV) ]`
	`| [< (@AP < PRELAT|PRELS|PRELS-SB|PRELS-OA`
	`|PRELS-DA|PWAT|PWS|PWAV) ] )`
	`>>  S|CS|DL & !> CS`

Table 2: Definitions of Syntactic Units Part I.

Clauses for English are characterized as structures that contain a subject and a finite verb (Hunt, 1965). Different from English, German allows subjectless sentences, so all maximal projections headed by a finite verb, as well

as elliptical constructions, where the finite verb is omitted, are considered as clauses.

Dependent clauses are embedded clauses. They fulfill a grammatical function in the clause that they are embedded in (see (Gallmann and Sitta, 2004, 120-133) for a detailed description). Dependent clauses can be further grouped into:

1. Relative Clauses
   Das ist das Paper, das ich lesen sollte. [That is the paper that I should read.]
2. Interrogative Clauses
   Er fragt, ob er das Paper lesen soll. [He asks whether he should read the paper.]
3. Conjunctional Clauses
   Sein Kollege sagt, dass er das Paper lesen soll. [His colleague says that he should read the paper.]

While many dependent clauses are headed by a subordinating conjunction or a pronoun there are also dependent clauses without such markers (*Er sagt, er hat keine Zeit. [He says he does not have the time.]*). Additionally there are "satzwertige Infinitive" - infinite clauses that have the same function as a dependent clause (Gallmann and Sitta, 2004; Bech, 1955; Meurers, 2000) (*Peter besucht einen Kurs, um Deutsch zu lernen. [Peter attends a course to learn German.]*).

T-Units are defined as "one main clause plus any subordinate clause or non clausal structure that is attached to or embedded in it" (Hunt 1970, p. 4; cf. Lu, 2010). Only independent clauses (including their dependents) count as a T-Unit.

At the phrasal level, *coordinated phrases* are defined as coordinated adjective, adverb, noun, and verb phrases. *Verb phrases* include non-finite as well as finite verb phrases. Finally, *complex nominals* are nouns with an adjective, possessive or prepositional phrase, relative clause, participle, or appositive. Nominal clauses, gerunds and infinitives in subject position are also included.

For the implementation these definitions had to be operationalized. We used *TregEx* (Levy and Andrew, 2006) to search parse trees with regular expressions. The tags in the operational definitions follow the *NEGRA* annotation scheme (Skut et al., 1997). Table 2 and Table 3 show the explicit definitions for all constructions that we extracted from parse trees, except for straightforward mappings such as: *verb phrase - VP*.

Unit	Definition

interrogative clause	`(@S`
	`[ <1 PWAT|PWS|PWAV]`
	`| [< (@AP < PWAT|PWS|PWAV) ]`
	`| [< (@NP < PWAT|PWS|PWAV) ] )`
	`>>  S|CS|DL & !> CS`

conjunctional clause	`(@S`
	`[ <1 KOUS|KOUI] )`
	`>>  S|CS|DL & !> CS`

dependent clause wo. subord. conj.	`(@S`
	`[<1 (VVFIN|VAFIN|VMFIN <<, !/\\?/)] . /,/`
	`|[<1 (@NP . VVFIN|VAFIN|VMFIN )] , /,/ )`
	`> S|CS|DL & !> CS`

'satzwertige Infnitive'	`@VP [< VZ , /,/] | [< VZ . /,/]`
dependent clause	dependent clause w. subord. conj + dependent clause wo. subord. conj + 'satzwertige Infnitive'
T-Unit	clause that is not (but may contain) a dependent clause
complex T-Unit	T-Unit that contains dependent clause
coordinated Phrase	`CAC|CAVP|CNP|CVP`
complex nominal	`@NP|NN|NE < S | < @AP` `| < @PP | < @PP | < ADJA|ORD`

Table 3: Definitions of Syntactic Units Part II.

3 Data

"Glückwunsch zur deine prüfung, Ich bin schietoll für dich, für mir, get mir gut und Ich finde sehr gut dass deine tante und onkel besucht in Istanbul hat, und was Ich wüsche in istanbul ist dass eine Mütze, Mütze finde ich schön."

Figure 1: Example for a text passage with spurious punctuation and spelling mistakes from an essay in the *MERLIN* corpus.

The data used in this theses consists of 1027 German essays from the *MERLIN* corpus[5]. *MERLIN* is a multilingual European project that aims at "creating a freely accessible multilingual online platform for the illustration of CEFR levels"(Wisniewski et al., 2011, 35). The texts were produced by language learners of different native language backgrounds during German as a second or foreign language exams. The learners were asked to write a short free text for a certain task[6], for example writing a letter to a friend or writing a fake job application. There were three different tasks per exam type. All essays were labeled with CEFR proficiency levels by trained human raters. The CEFR levels comprise six proficiency levels that are mainly defined on a functional scale: A1 (Breakthrough), A2 (Waystage), B1 (Threshold), B2 (Vantage), C1 (Effective Operational Proficiency), C2 (Mastery). Level C2 was not represented in our data set.

In the course of the *MERLIN* project, the learner essays were digitalized and stored using the Paula XML annotation scheme (Wisniewski et al., 2011). The annotation of the data is still in progress. We used a plain text version of the original essays[7].

There are roughly 200 texts for each exam type. However, it is the essay rating level or proficiency level that reflects the actual language mastery demonstrated in an essay. The distribution among essay rating levels is not even. There are only 57 texts that received the rating A1 and only 75 that were rated with C1. Figure 2 shows the distribution of texts across the essay rating levels. As a first superficial statistical analysis of the data showed, there is a high correlation between exam type and text length ($r = 0.86$) and essay rating level and text length ($r = 0.84$). This is illustrated in Figure 3 and Figure 4. As a result much less data is available for the low essay rating levels, especially for A1.

[5]German is just one of several European languages included in *MERLIN*

[6]Information on the task was gathered from email correspondence with Detmar Meurers and Serhiy Bykh

[7]Thanks to Serhiy Bykh and Adriane Boyd for making the plain text version available

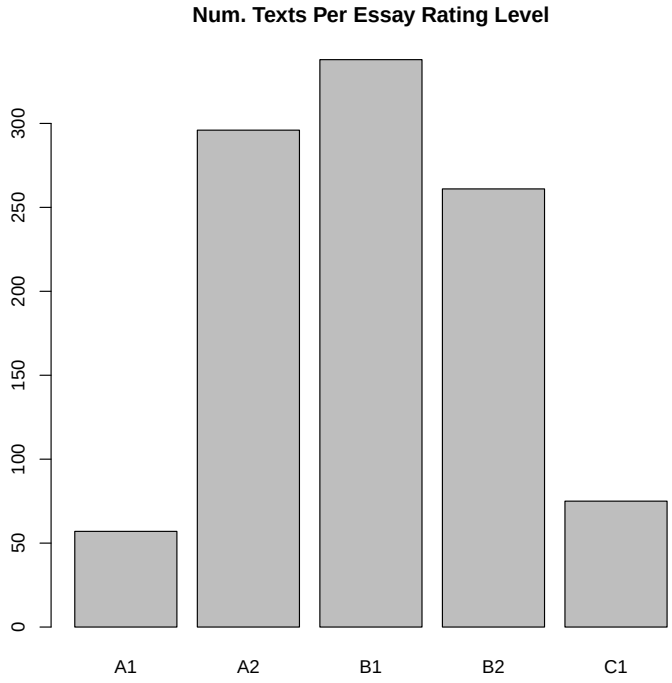

Figure 2: Number of essays per essay rating level.

The dataset poses several challenges. The skewed distribution of data across the essay rating levels can complicate machine learning. There are various possible factors that could also influence classification: different text lengths, different L1 backgrounds and different tasks for essay writing.

Additionally the data is very noisy compared to 'standard' written language such as newswire. Although nonstandard language use and mistakes can be utilized for classifying learner language, they first and foremost pose a challenge. Passages like the one shown in Figure 1 are not uncommon in the *MERLIN* data. Punctuation is often spurious. Spelling and word order mistakes are frequent. This is problematic for NLP-Tools. Figure 5 shows the parse tree of the above text passage. Some words were tagged with the wrong part-of-speech. More strikingly, the whole passage was analyzed as a single sentence because there is only one sentence final punctuation mark.

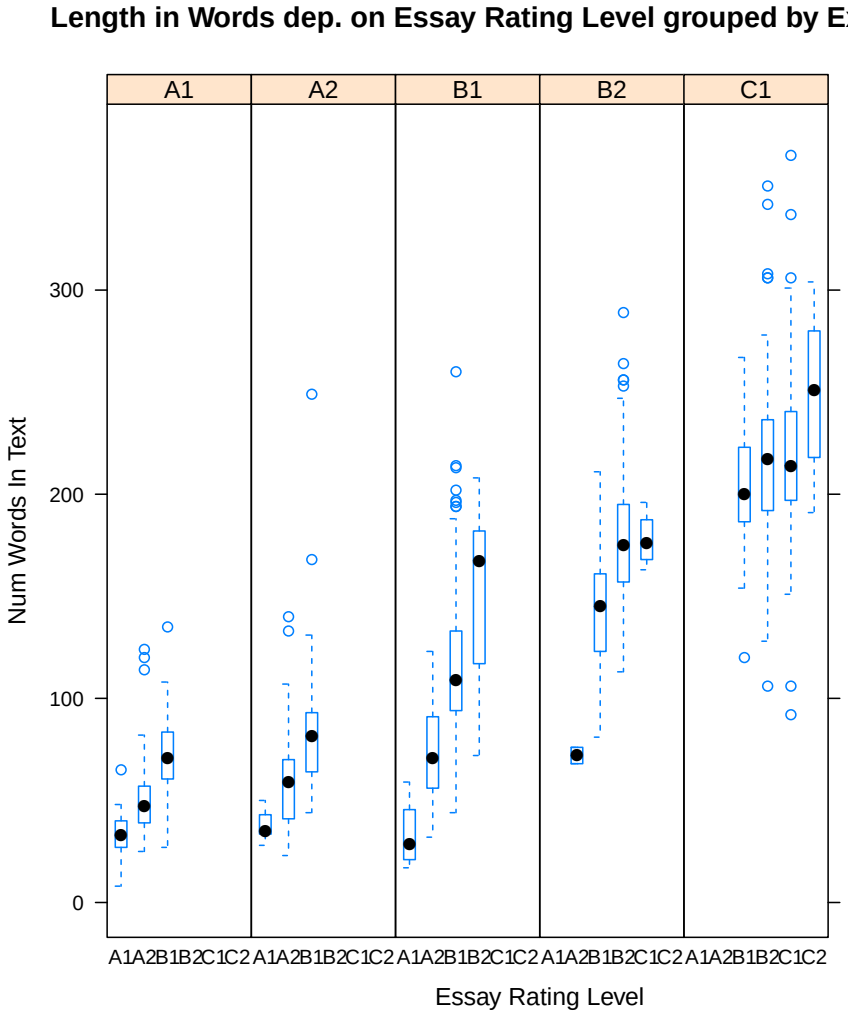

Figure 3: Text length in words depending on essay rating level grouped by exam type.

4 Preprocessing

4.1 Components of the Preprocessing Pipeline

We used a plain text version of the 1027 German original learner texts from the *MERLIN* corpus. Due to the nature of the essay writing tasks, some texts contained letterheads. The letterheads were tagged in the plain text version. For the analysis they were removed as they don't contribute to the proficiency analysis and are problematic for some tools, such as sentence detectors and parsers.

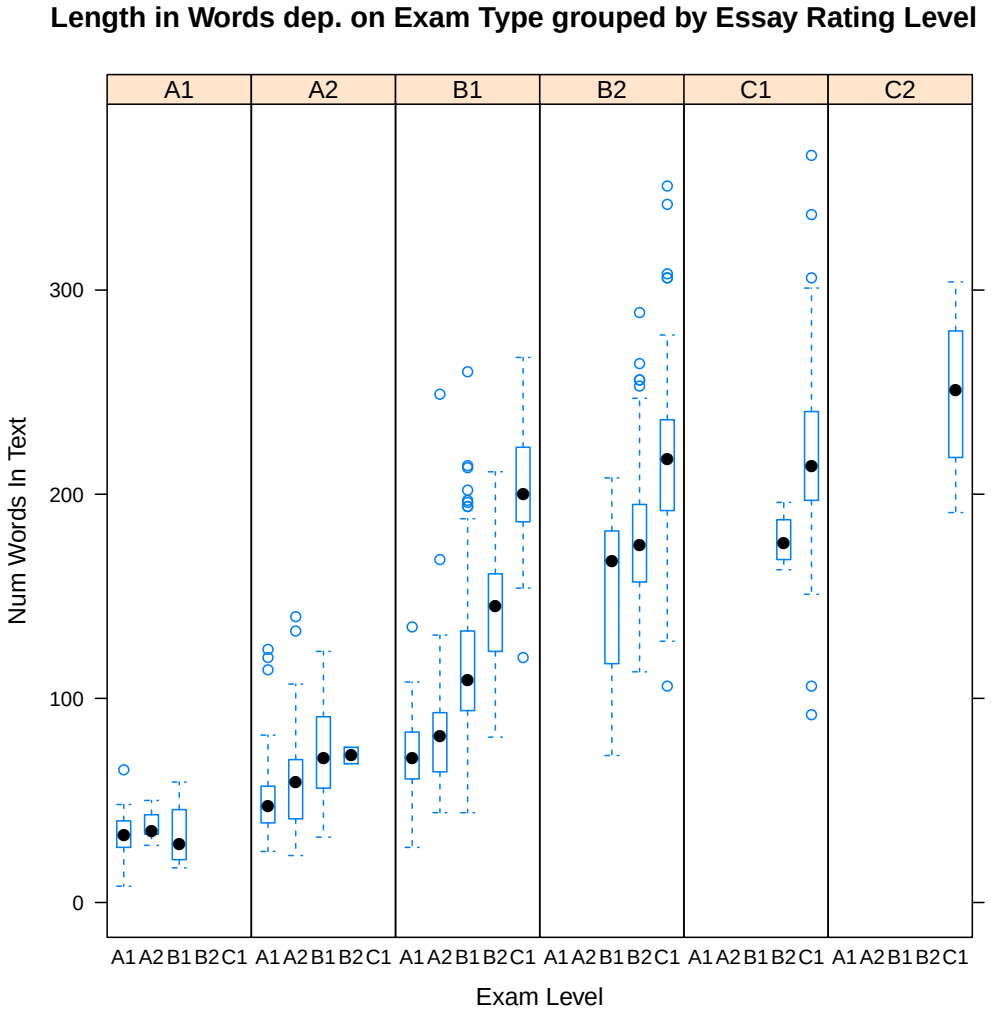

Figure 4: Text length in words depending on exam type grouped by essay rating level.

The spurious punctuation in some of the *MERLIN* essays prompted us to build a tool for missing sentence boundary detection. Although the results were promising, the tool was not reliable enough to be used in this work. Still, the approach is discussed in Section 4.2 and might be further developed in the future. For now, the *OpenNLP*[8] *SentenceDetectorME* (OpenNLP Tools 1.5) was used for sentence segmentation. A pre-trained model for German is available at *OpenNLP*. A number of problems occurred when using the tool on our data.

[8] `http://opennlp.apache.org/`

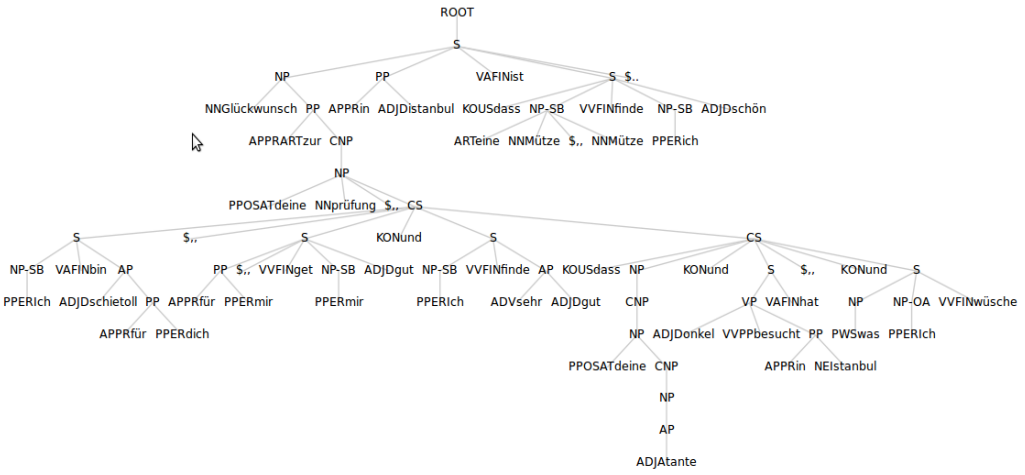

Figure 5: Example for a parsed text passage with spurious punctuation and several spelling mistakes from an essay in the *MERLIN* corpus. Visualized with ProD (Culy et al., 2012)

1. If there were several sentence final punctuation marks, the sentences were not split.
2. Abbreviations were often not handled correctly.

1. was addressed by finding all instances of multiple sentence final punctuation marks using the regular expression `[!?\\.]{2,}` and replacing them by a single full stop. For handling 2. a custom model was created for *SentenceDetectorME*. *OpenNLP* offers a command line interface for sentence detector training[9]. In addition to custom training data it is possible to specify a list of abbreviations for improving the tool's performance. For that purpose a list of German abbreviations was mined from the web[10]. *SentenceDetectorME* was then trained on *NEGRA* 2 Corpus Export[11] and the additional abbreviation list.

For tokenization *OpenNLP TokenizerME* (OpenNLP Tools 1.5) was used. The model was trained on the same data and abbreviation list as the sentence detector to maintain compatibility.

To reduce problems caused by spelling errors, a Java API for *Google Spell Check* (version 1.1)[12] was used. When submitting a word or text for spell correction, the language can be chosen[13]. When sending a request, multiple corrected suggestions are returned for each misspelled word, ranked according to the spell checker's confidence. The suggestion with the highest confidence was picked.

The spell corrected data was tagged using a Java interface (version 0.0.8)[14] to the *RFTagger* (Schmid and Laws, 2008), a statistical tagger that provides a fine grained morphological analysis. As some of the further processing steps relied on *Stuttgart-Tübingen Tagset*, all tags were additionally converted to this tagset, using the converter class that is integrated into the Java interface.

RFTagger lemmatizes words with a Perl script that looks up lemmas in a lexicon. This capability, however, is not included in the Java interface for *RFTagger*. An attempt to use the scripts for lemmatization separately made the application extremely slow. Therefore we eventually used *TreeTagger*

[9] `http://opennlp.apache.org/documentation/1.5.2-incubating/manual/opennlp.html#tools.sentdetect.detection`

[10] `http://german.about.com/library/blabbrev.htm`

[11] `http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html`

[12] `https://code.google.com/p/google-api-spelling-java/`

[13] There were some encoding issues, when using German. The Java encoding must be set to UTF-8, however, the correction suggestions for German are returned ISO-8859-1 encoded, so they have to be coded to UTF-8 again. Also, the rate of making requests has to be reduced in order to not be blocked by Google.

[14] `http://www.sfs.uni-tuebingen.de/~nott/rftj-public/`

(Schmid, 1995) with a Java wrapper by Richard Eckart de Castilho[15] for lemmatization. The problem with this approach is that *TreeTagger* and *RFTagger* don't necessarily assign the equivalent part-of-speech tag to a word. However, after supervising this potential problem during a preprocessing run, it turned out that the differences were minimal.

All documents were parsed with the *Stanford Parser* for German, a lexicalized probabilistic context free grammar parser. We used the standard model for German (version 2.0.4) (Rafferty and Manning, 2008) trained on the *NEGRA* Corpus[16]. As we wanted to experiment with dependency based features as well, all documents were additionally parsed with *MATE* dependency parser (Bohnet, 2010), using the standard model for German (Seeker and Kuhn, 2012), which was trained on the *TIGER* Corpus.

For storing annotated documents, the serializable class `AnnotatedDocument` was implemented. The advantage is that all annotation layers can be stored and accessed in a principle way.

4.2 Sentence Boundary Detection for Essays with Missing or Spurious Punctuation

Most conventional approaches address sentence segmentation as a problem of punctuation disambiguation. Machine learners are trained with features such as part-of-speech frequency (Palmer and Hearst, 1997) and lexical information (Reynar and Ratnaparkhi, 1997) in the context of the candidate punctuation mark and obtain high accuracies (98%, (Reynar and Ratnaparkhi, 1997)). Since sentence final punctuation is sometimes spurious in the *MERLIN* essays, those tools do not work reliably. It would be most desirable to find a solution to sentence boundary detection that does not rely on punctuation.

4.2.1 Survey

Finding sentence boundaries in the absence of punctuation is also a relevant task in speech recognition. Many approaches (Stolcke and Shriberg, 1996; Gotoh and Renals, 2000; Liu et al., 2004) use a special type of ngram model introduced by Stolcke and Shriberg (1996). To incorporate sentence boundary probabilities into ngram models, Stolcke and Shriberg (1996) postulated a possible sentence boundary tag after each word. Hidden event models were trained with these ngrams to find the most likely positions for sentence

[15] `http://code.google.com/p/tt4j/`

[16] `http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html`

boundaries. Later approaches often combined this technique with prosodic information (Liu et al., 2004; Liu et al.; Favre et al., 2008). Favre et al. (2008) tried to include syntactic clues by using parser probabilities of text spans.

After reviewing approaches to sentence boundary detection, that were used in speech recognition, it seems doubtful that the same techniques could be applied to our data without much adjustment. There is no prosodic information in the written texts that could enhance the performance of the hidden event ngram models. Also, in contrast to the transcript of a speech recognizer, the learner essays mostly do contain some punctuation marks. As a first step, the passages where punctuation is missing would have to be identified.

Nagata et al. (2010) describe an approach to detecting missing sentence boundaries in texts produced by Japanese learners of English. They put forward a method for automatically generating training data for missing sentence boundary detection and then used this data to train Support Vector Machines with features like sentence length probability, number of words beginning with a capital letter, number of verbs, prepositions, wh-words, conjunctions, and frequencies of non-sentence final punctuation marks. Their best model achieved promising result (0.716 % F-measure).

We conclude that implementing a component for placing missing punctuation marks in the absence of punctuation is not feasible in this thesis. Nevertheless we attempted the first step by trying to automatically identify missing sentence boundaries using a similar approach as Nagata et al. (2010). Such a tool would be useful to flag essays with spurious punctuation. Those essays could then either be excluded or manually corrected.

4.2.2 Building a Missing Sentence Boundary Detector

This section describes the implementation of a missing sentence boundary detector. Following Nagata et al. (2010) we automatically generated training data from a corpus of learner language. A classifier was then trained with linguistic features that are promising indicators for missing sentence boundaries.

4.2.2.1 Creating Training and Test Set We automatically created training data from the *Kobalt-DAF* (Zinsmeister et al., 2011) data. *Kobalt-DAF* is an ongoing project that researches the linguistic properties of texts written by learners of German from different L1 backgrounds. The current version of the corpus comprises of 69 texts. We followed the procedure

proposed by Nagata et al. (2010). Each training sample consists of one sentence. To obtain samples, the *Kobalt-DaF* texts were sentence split with the OpenNLP sentence segmenter as described in Section 4.1. The resulting sentences were used as negative training instances. To create positive instances, pairs of sentences were concatenated. The order of the original sentences was shuffled beforehand, to avoid that contiguous sentences were combined.

Two additional test sets were created for evaluating the sentence boundary detector on two different essay rating levels (A2 and B1) from the *MERLIN* corpus. 1000 sample sentences as segmented by OpenNLP sentence segmenter (section 4.1) were manually labeled for essay rating level A2 and B1 respectively (henceforth MA2 and MB1). Among the 1000 sentences for each essay rating level there were only 61 positive instances in MA2 and 36 in MB1. This uneven distribution of negative and positive cases has to be kept in mind when evaluating the classifiers.

4.2.2.2 Preprocessing for Sentence Boundary Detection The preprocessing procedure was very similar to the steps described in section 4.1. Minor difference arose from the fact that for some problems, better solutions were found later on. Because of the need to manually annotate sentences, which is very time consuming, the experiments could not be repeated again at a later point.

No version of the data with tagged letterheads had been available at the time when we built the missing sentence boundary detector. To exclude most of the potential letterheads, only lines with more then two tokens were considered. A full stop was appended to lines that did not yet have a sentence final punctuation mark. The fact that OpenNLP sentence segmenter has problems with multiple punctuation marks *!!!* had not been addressed.

Spell checking was only implemented later. Fine grained tags such as those delivered by *RFTagger* were not needed for the task at hand, so the *OpenNLP Tagger* was used with the standard model provided by OpenNLP. The data was parsed with the *Stanford Parser* for German.

4.2.2.3 Features We adapted most of the features used by Nagata et al. (2010). Nagata et al. (2010)'s feature set included the number of capitalized words that are neither proper nouns nor the pronoun *I*. In German, however, besides from the proper nouns, also common nouns are capitalized, so the number of capitalized words excluding all nouns was counted. Nagata et al. (2010, 1272) argued that verbs are indicative of missing sentence boundaries, as a sentence is assumed to consist of a single verbs unless "it contains conjunctions and/or clauses". Therefore they counted the verbs

excluding present and past participles. We implemented two distinct verb features instead: the number of verbs per word, and the number of finite verbs per word. To include the presence of conjunctions as indicators Nagata et al. (2010) considered the frequencies of different conjunctions as well as the total number of conjunctions. In this work only the total number of conjunctions is used as a feature. Similarly, we only used the total number of wh-Words and prepositions.

Furthermore we included the following features into our approach: frequency of colons, semicolons, commas, total number of punctuation marks, and sentence length probability. Sentence length probability is the likelihood that a sentence is of a particular length, measured on the distribution of already observed sentence lengths. For computing this probability Nagata et al. (2010) assumed that sentence length has a Gaussian distribution and used the probability density function. The observed sentence lengths can be computed from the training data. In addition to this more sophisticated measure of sentence length, we also included sentence length as the number of words per sentence. It was used as a baseline for all experiments.

Additionally three new features were examined that rely on information from parse trees. The number of clauses per number of words seemed to be a promising indicator, since the parser mostly recognized clauses in spite of missing punctuation. This resulted in parse trees which contained many clauses under one ROOT node. As the parser often related these clauses to each other by using coordination, coordinated clauses also seemed a good indicator. Finally parse tree height seemed promising as it could help to distinguish sentences which have many clauses because of heavy embedding from relatively flat trees, that contain many clauses because of missing sentence boundaries.

4.2.2.4 Experiments First the performance of three different machine learning algorithms - Naive Bayes, Sequential Minimal Optimization Algorithm (SMO) and Decision Tree (J48) - were compared. All algorithms were used with their standard configurations from *WEKA*. We used 10 fold cross validation and the whole feature set to build and test the models. We report precision, recall and F-measure for the positive class (missing sentence boundary)[17]. The experiments showed that the Decision Tree Algorithm was the best choice.

We build different models on the generated training data using 10 fold

[17]Cost sensitive learning could have been applied to the sentence boundary detection task. However, as the main task was proficiency level assessment, it seemed unwise to spend too much time on sentence boundary detection.

Name	Formula
Avg. Num. Verbs Per Words	# verbs / # W
Avg Num. Finite Verbs Per Words	# finite verbs / # W
Avg Num. Conjunctions Per Words	#conjunctions / # W
Avg Num. wh-Words Per Words	# wh-words / # W
Avg Num. Prepositions Per Words	# prepositions / # W
Avg Num. Commas Per Words	# commas / # W
Avg Num. Colons Per Words	# colons / # W
Avg Num. Semicolons Per Words	# semicolons / # W
Avg Num. Capital Not Nouns Per Words	# capital not nouns / # W
Avg Num. Punctuation Per Words	# punctuation marks / # W
Sentence Length Probability	
Num. Tokens In Sentence	# W
Avg Num. S Nodes Per Words	# clauses / # W
Avg Num. CS Nodes Per Words	# coordinated clauses / # W
Parse Tree Height To Words Ratio	parse tree height / # W

Table 4: Features for missing sentence boundary detection.

Classifier	Accuracy	Precision	Recall
SMO	92.67	95	90
Naive Bayes	75.07	70	90
Decision Tree J48	94.56	97	92

Table 5: Comparison of different classifier algorithms

cross validation. Then each of these models was evaluated on the hand labelled MA2 and MB1 test sets. A model build with all features (ALL) was compared to a sentence length baseline (BL). As indicated by Nagata et al. (2010), the capitalization feature was responsible for many false positives in their experiments. Therefore we created a model without the capitalization feature (ALL-CAP).

A detailed summary of the results can be seen in Table 6. When tested with cross validation ALL performed better than BL. However, when tested on the MA2 and MB1 sets, ALL's results dropped below those of BL. Especially the precision plummeted. For ALL-CAP, the F-measure dropped when evaluated with cross validation, but increased when tested on MA2 and MB1. The reason is probably, that the spelling conventions regarding capitalization are complied with to a different degree in the training data and the *MERLIN* test sets. Generally the performance on MA2 and MB1 was not too impressive. Even the ALL-CAP model only slightly exceeded the baseline for MA2 and the performed below the baseline for MB2.

Dataset	Model	Accuracy	F-Measure	Precision	Recall
Train	BL	79.00	80	77	83
	ALL	94.56	94	97	92
	ALL-CAP	88.90	88	93	85
MA2	BL	93.90	38	50	31
	ALL	57.56	16	9	64
	ALL-CAP	91.10	40	34	42
MB1	BL	92.10	60	25	61
	ALL	63.53	12	6	66.7
	ALL-CAP	88.48	24	16	50

Table 6: Comparison of different models

4.2.2.5 Conclusion Although the proposed method seemed promising when tested with cross validation, the results on the *MERLIN* test sets were not satisfactory. The immensely skewed distribution of positive and negative samples and too little similarity between the training and test data might have contributed to this result. Additionally it might be hard to find one model that can be successfully applied to all levels of the *MERLIN* corpus, as the essays at different levels are so dissimilar. It is concluded that the missing sentence boundary detection component is not yet mature enough to be used in the further course of this work. It is however considered worthwhile to continue its development in the future.

5 Features For Language Proficiency Assessment

In this section we introduce our feature set for language proficiency classification. We implemented a variety of syntactic, lexical, morphological, and language model features that were informed by different fields of research.

5.1 Lexical Features

A number of lexical indices of language proficiency have emerged from research on language acquisition, and have recently also been deployed in readability assessment and language proficiency classification (Vajjala and Meurers, 2012; Hancke et al., 2012; Vajjala and Loo, 2013). We integrated a wide range of lexical indices into our approach to examine their usefulness for German proficiency assessment.

5.1.1 Lexical Diversity Measures

Lexical Diversity. Lexical diversity or lexical richness designates the variety and range of the vocabulary used in a text or speech (McCarthy and Jarvis, 2007). Measures of lexical diversity have been widely deployed as indices for e.g., vocabulary knowledge and writing quality. They have also been used to measure the lexical proficiency of language learners (Crossley et al., 2011a; Lu, 2012; Graesser et al., 2004).

The most commonly used lexical diversity index is probably Type-Token Ratio (TTR) (McCarthy and Jarvis, 2007). It is known that TTR depends on text length. McCarthy and Jarvis (2007) explained this flaw with Heaps law ((Heaps, 1978) c.f. McCarthy and Jarvis (2007)): When a text becomes increasingly long (many tokens), the chance that new words (types) occur becomes more unlikely. As a result, long texts appear to be less lexically diverse when measured with TTR. In a comprehensive study McCarthy and Jarvis (2007) examined a number of TTR variants that were developed for decreasing the dependency on text length - mostly involving some mathematical transformation. These variations include Root Type-Token Ratio, Corrected Type-Token Ratio, Uber Index and Yule's K. McCarthy and Jarvis (2007)'s results showed that none of these measures were independent of text length, although for Yule's K and Uber Index the effect was minor[18].

More recently vodc (MacWhinney, 2000) has been proposed as a new index of lexical diversity. It tries to overcome the dependency on text length by repeatedly drawing samples of different sizes from a text, and calculating the mean TTR of this samples. A formula using the D coefficient (Malvern et al., 2004, c.f. McCarthy and Jarvis (2010)) is then deployed to fit a theoretical curve to the empirical TTR curve obtained from the random samples (McCarthy and Jarvis, 2007, 2010).

The results of McCarthy and Jarvis (2007)'s examination suggested that vocd is not independent of text length. Furthermore they showed that deploying the hypergeometric distribution[19] can replace the curve fitting procedure in vocd. The hypergeometric distribution describes the probability of success for drawing (without replacement) a particular number of tokens of a particular type from a sample of a certain size (McCarthy and Jarvis, 2007, 2010). McCarthy and Jarvis (2007) proposed HDD, an index of lexical diversity, that is very similar to vocd, but makes use of the hypergeometric distribution directly.

McCarthy and Jarvis (2010) introduced the measure of textual lexical

[18] According to McCarthy and Jarvis (2007) 2% for Yule' K and 3% for Uber Index of the variance was explained by text length.

[19] A probabilistic distribution

diversity (MTLD). It is a sequential approach to measuring lexical diversity by calculating the mean length of a string sequence, that maintains a default TTR value. Whenever a new type is found, the TTR is calculated. As soon as the default value is reached, a factor count is increased and the TTR is reset. The same procedure is repeated from that point in the text to the end of the text. If the last portion of text does not reach the default TTR, a partial factor is calculated[20]. MTLD is then calculated by dividing the number of tokens by the number of factors. The same procedure is repeated beginning at the end of the text. The final score is the mean of the forward and backward MTLD value (McCarthy and Jarvis, 2010).

Measures using type to token ratio	Formulas
Type-Token Ratio	$\#Typ/\#Tok$
Root Type-Token Ratio	$\sqrt{\#Typ/\#Tok}$
Corrected Type-Token Ratio	$\sqrt{\#Typ/(2 * \#Tok)}$
Bilogarithmic Type-Token Ratio	$\log(\#Typ)/\log(\#Tok)$
Uber Index	$\log(\#Tok)^2/\log(\#Typ/\#Tok)$
Yule's K	$10^4 * (sum(fX * X^2) - \#Tok)/(\#Tok^2)$, X=vector of freq. for each type, fX=frequency of each type freq. in X
HD-D	McCarthy and Jarvis (2007)
Measure of Textual Lexical Diversity	McCarthy and Jarvis (2010)

Table 7: *Type-Token Ratio* and variations

Lexical Density and Variation. Lexical density (originally Ure (1971), c.f. Lu (2012)) and lexical variation measures are closely related to the type to token based ratios that we discussed in the previous section. Lexical Density measures the ratio of lexical words (or 'content words') to all words. Lexical variation measures include different ratios that measure the variation within specific syntactic categories, for example Verb Variation (verb types to verb tokens) and Noun Variation (noun types to noun tokens). Lu (2012) deployed lexical density and variation measures in combination with lexical diversity measures to examine the relationship between lexical richness and the quality of English learner productions. Those measures also have been proven to be good indicators of readability by Vajjala and Meurers (2012) for English texts and Hancke et al. (2012) for German texts. However, although these measures have been successfully applied in a number of approaches, doubts remain that - like Type-Token Ratio - they might depend on text length.

[20] $(1 - ttr_{leftover})/(1 - ttr_{default})$

Measures using lexical type distributions	Formulas
Lexical Density	$\#Tok\_Lex/\#Tok$
Lexical Word Variation	$\#Typ\_Lex/\#Tok\_Lex$
Noun Variation	$\#Typ\_Noun/\#Tok\_Lex$
Adjective Variation	$\#Typ\_Adj/\#Tok\_Lex$
Adverb Variation	$\#Typ\_Adv/\#Tok\_Lex$
Modifier Variation	$(\#Typ\_Adj + \#Typ\_Adv)/\#Tok\_Lex$
Verb Variation 1	$\#Typ\_Verb/\#Tok\_Verb$
Verb Variation 2	$\#Typ\_Verb/\#Tok\_Lex$
Squared Verb Variation 1	$\#Typ\_Verb^2/\#Tok\_Verb$
Corrected Verb Variation 1	$\#Typ\_Verb/\sqrt{2\#Tok\_Verb}$
Verb Token Ratio	$\#Tok\_Verb/\#Tok$
Noun Token Ratio	$\#Tok\_Noun/\#Tok$
Verb-Noun Token Ratio	$\#Tok\_Verb/T\#ok\_Noun$

Table 8: Lexical variation features

A comprehensive set of lexical diversity, density and variation measures was included as can be seen in Tables 7 and 8.

5.1.2 Depth of Knowledge Features

As suggested by Crossley et al. (2011b), not only vocabulary size and diversity should be taken into account, but also the depth of lexical knowledge. Their results showed that lexical frequencies and hypernym scores are promising features for proficiency assessment.

Lexical Frequency. The general frequencies of the lexical items used by a learner are often seen as an indicator of lexical proficiency. As summarized by Crossley et al. (2011b), it is assumed that learners with a higher proficiency use less frequent lexical items. High word frequency leads to more exposure, which makes a word easier to learn (Ellis (2002), cf. Crossley et al. (2011b)). This hypothesis is supported by studies of lexical acquisition (Balota and Chumbly (1984), Kirsner (1994) c.f. Crossley et al. (2011b)).

According to Crossley et al. (2011b) the correlation between word frequency and acquisition has also been confirmed for lexical development in second language learning: In most studies it was shown that beginning learners were more likely to use and comprehend high frequency lexical items (Crossley and Salsbury (2010); Ellis (2002) c.f. Crossley et al. (2011b)). To measure the frequencies of lexical items , Crossley et al. (2011b) retrieved content word frequency scores from the *CELEX* database (Baayen et al., 1995)

- a psycholinguistic database that includes word frequencies as counted from a reference corpus.

We extracted content word frequencies from *dlexDB* (Heister et al., 2011). *DlexDB* frequency scores are based on the "Kernkorpus des Digitalen Wörterbuchs der deutschen Sprache" (*DWDS*). The *DWDS* corpus comprises of 122.816.010 tokens and 2.224.542 types. For providing relevant annotations such as part-of-speech tags, lemmas or syllables, *dlexDB* relied on automatic annotation with NLP tools (Heister et al., 2011). *DlexDB* was chosen over *CELEX* because it is more recent and based on a larger reference corpus (6 million vs. 100 million running words).

DlexDB offers a number of interesting scores for each word, such as absolute and normalized frequencies and log absolute and normalized frequencies for annotated types (type, word pair), types, lemmas and ngrams, but also familiarity scores and frequency ranks (see Heister et al. (2011) for a comprehensive list). Currently the API for fully automatic queries is still under development[21]. However, the database can be queried via a web interface[22]. Queries can be send for individual lexical items or for lists of items, and the results can be exported after creating a user account (up to 10 000 results).

For making all the necessary frequencies from *dlexDB* available to our tool, a custom *dlexDB* version with all the words in the *MERLIN* corpus and all the scores that were relevant had to be created by manually querying *dlexDB*. A list with all spell checked words in the *MERLIN* corpus was created. It was split into tiles of not more than 4000 words, in order to remain below the export limit. Each tile was then used to query *dlexDB* manually over the web interface: For each word the following scores were retrieved:

1. Absolute Annotated Type Frequency (ATF)
2. Log absolute Annotated Type Frequency (LATF)
3. Absolute Type Frequency (TF)
4. Log absolute Type Frequency (LTF)
5. Absolute Lemma Frequency (LF)
6. Log absolute Lemma Frequency (LLF)

Finally all non-matches and all words that did not match the category of 'content word' (incl. modals) were cleaned out.

Features that measure the general frequency of the lexical items used in an essay were created using the scores 1.-6. All types in an essay were looked

[21] personal correspondence with *dlexDB* developers

[22] `http://www.dlexdb.de/query/kern/typposlem/`

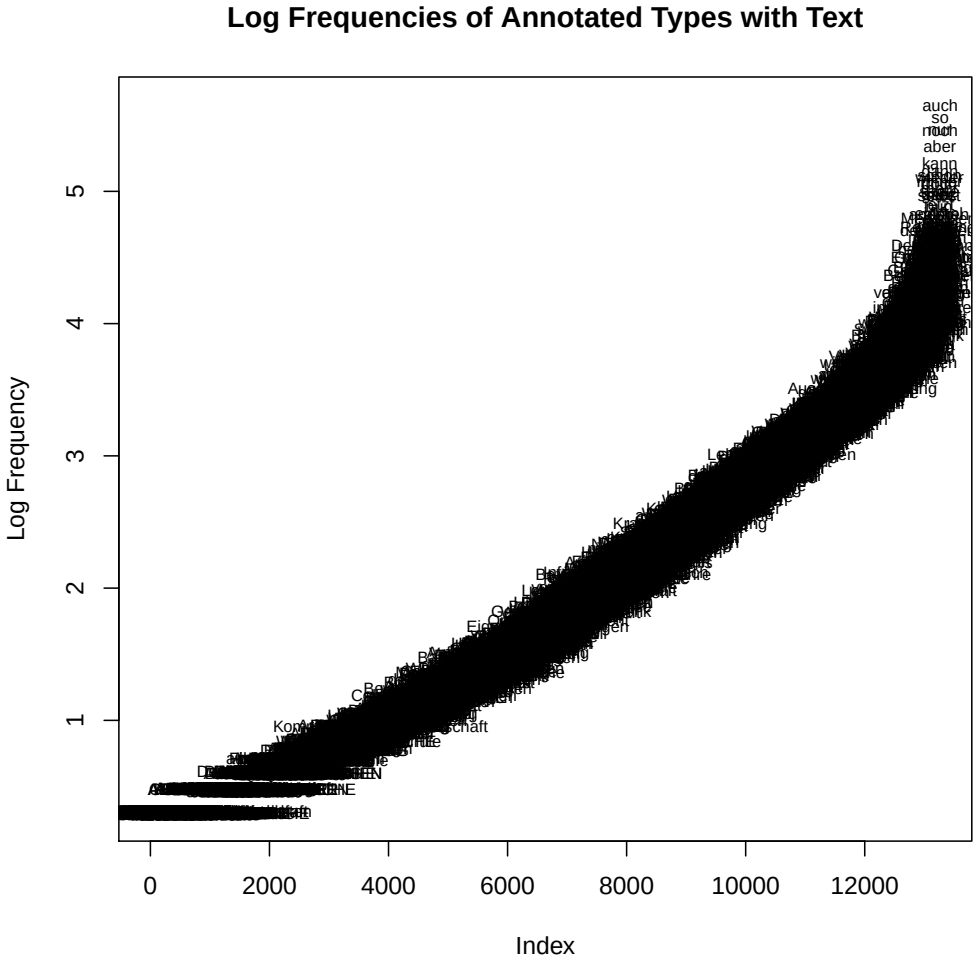

Figure 6: Log Annotated Type Frequencies.

up in the custom *dlexDB*. For each type of frequency score (ATF, .. , LLF) the scores of all types in an essay were summed up. They were then divided by the total number of types that had been found in the custom *dlexDB* (M = matches). We used the matches as a denominator in the ratios rather than the number of types in the essay: the more types are found in dlexDB, the more items add to the sum of scores. Hence creating ratios with all types in an essay as a denominator would not be fair, since they do not reflect the number of types that actually contributed to the sum of scores.

Additionally we calculated the ratio of words occurring in a specific frequency band of LATF. To decide on frequency bands, the LATFs of the whole custom dlexDB were plotted using R (Figure 6). It can be seen that

dlexDB Frequency Features	Formulas
Annotated Type Ratio	sum(ATF) / # M
Type Ratio	sum(TF) / # M
Lemma Ratio	sum(LF) / #M
Log Annotated Type Ratio	sum(LATF) / # M
Log Type Ratio	sum(LTF) / # M
Log Lemma Ratio	sum(LLF) / # M
Ratio of Words In Log Frequency Band	$\text{sum}(\text{LATF}_{\text{inFrequencyBandN}}$ / # M
Ratio of Lex. Types not in Dlex	# not M / $\#Typ_{Lex}$

Table 9: Lexical frequency features using *dlexDB*

there are many words with a log frequency between zero and one. Above one, a relatively linear curve ascends up to four, and a steep rise follows up to five. Between five and six the items become more sparse. The types are mainly adverbs or modals. Consequently six frequency band were assumed, one for each step from zero to six. The ratio of all lexical types that were not found in *dlexDB* was included as a measure of orthographical or lexical errors. A summary of all frequency features can be seen in Table 9.

Lexical Relatedness. Conceptual relations between lexical items are important for the organization of lexical knowledge in the human mind. Hypernymy is a relation of more specific terms (hyponyms - subordinate words) to less specific terms (hypernyms - superordinate words) (Crossley et al., 2009). Hypernyms can have a grouping function or be an umbrella term for hyponyms (e.g., flower is an umbrella term for roses, lilies, daisies ...). Crossley et al. (2009) claimed that hypernymic relations are good indicators for lexical organization and the depth of lexical knowledge. The results of their study indicated that in second language learning, the number of hypernymic relations increases as the learner makes progress. Crossley et al. (2009, 2011b) used *WordNet*[23] (Miller, 1995) to retrieve hypernymy scores. *WordNet*'s organization is hierarchical: Hypernymy scores, that measure the distance to the root node, are provided. More abstract words have lower scores (Crossley et al., 2009).

Polysemous words have several related senses (Crossley et al., 2011b). Polysemy plays a role in the conceptual organization of lexical knowledge because overlapping word senses relate concepts to each other, and can thus form connections between lexical items (Crossley et al., 2010). In a longitudinal study of second language learners, Crossley et al. (2010) retrieved information about polysemy from *WordNet* synsets (groups of related lexical

[23] `http://wordnet.princeton.edu/`

items) to measure "the production of words with multiple senses over time". They found out that there is a correlation between growing lexical proficiency and the production of multiple word senses.

In this work *GermaNet* 7.0 was used to retrieve information about lexical relatedness. *GermaNet* is a machine readable lexical-semantic resource for German that is similar to *WordNet* for English. It is structured as a net or graph of concepts that are interlinked by semantic relations. Concepts are organized in synsets. Synsets are sets of lexical units with a closely related meaning. Most words belong to more than one synset. *GermaNet* also offers information about conceptual relations such as hypernymy or part-whole relations (Henrich and Hinrichs, 2010). *GermaNet* 7.0 consists of 74612 synsets and covers 99523 lexical units[24]. It is free for academic use after signing a license agreement, and a Java API[25] is available.

Six lexical relatedness features were implemented with information from *GermaNet*. The lexicon was queried with the lemma and word category of each noun and verb type in an essay. For all matches M (= lemma and word category combinations that were found in *GermaNet*) all synsets (SynS) were retrieved. All retrieved synsets were considered, as word sense disambiguation could not be implemented in the scope of this work. For each synset all immediate hypernyms and hyponyms were retrieved - that is hypernyms and hyponyms with a node distance of 1 from the synset.

The general connectedness of a word was measured by the average number of relations per synset that a word belongs to. The average number of hypernyms per match and average number of hyponyms per match measure the number of immediate super- and subordinate terms of a match, and are another index for a word's connectedness.

Polysemy measures were based on synset scores. The average number or word senses per word was calculated by using the number of synsets each match belonged to (= number of word senses). Additionally, the number of lexical units belonging to each of the synsets might be interesting. Synsets that contain more lexical units establish more relations and connections in the lexical network. Additionally the number of frames per verb found in *GermaNet* (VM) was included as a feature. *GermaNet* frames encode the subcategorization information of a verb. Table 10 shows a summary of all lexical connectedness features.

[24]Info on *GermaNet* 7.0 on the project website `http://www.sfs.uni-tuebingen.de/lsd/index.shtml`

[25]Here: Java API 7.0 `http://www.sfs.uni-tuebingen.de/lsd/tools.shtml`

Lexical Connectedness Features	Formulas
Avg. Num. Synsets per Match	# SynS / # M
Avg. Num. Lexical Units per Synset	sum(# LexUnit in SynS) / # SynS
Avg. Num. Relations per Synset	sum(# RelSynS of SynS)/ # SynS
Avg. Num. Hypernyms per Match	sum(# Hyper per Syns)/ # M
Avg. Num. Hyponyms per Match	sum(# Hypo per Syns) / # M
Avg. Num. Frames per Verb	# Frames / # VerbalM

Table 10: Lexical features measuring the connectedness of lexical items.

5.1.3 Shallow Measures and Error Measures.

As two shallow measures of lexical complexity, syllable count[26] and word length in characters were included. Those measures were used in language acquisition studies (Crossley et al., 2011a), but were also part of traditional readability formulas (Kincaid et al., 1975). Finally, errors on the word level were measured by counting the spelling errors found by *Google Spell Check*.

Other Lexical Measures	Formulas
Text Length	# Tok
Avg. Num. Characters per Word	# Char / # Tok
Avg. Num. Syllables per Word	# Syll / # Tok
Google Spell Check Error Rate	# Spell Erros / # Tok

Table 11: Shallow measures of lexical proficiency.

5.2 Language Model Features

Ngram language models can predict the probability of a specific sequence of words based on its history (Schwarm and Ostendorf, 2005).

$$P(w) = P(w_1)P(w_2|w_1)\prod_{i=3}^{m} P(w_i|w_{i-1}, w_{i-2}) \quad (2)$$

Language models can be deployed to measure textual complexity, as the distribution of word sequences tends to differ with complexity. Therefore they have been widely used in readability assessment (e.g., Schwarm and Ostendorf (2005); Petersen and Ostendorf (2009); Feng (2010)). Ngram frequency

[26]Syllable counts were obtained by simple heuristics: In German, a syllable mostly contains exactly one vowel - phonetically speaking. This means that in written language diphthongs, double vowels and the combination *vowel+e*, which indicates a long vowel, have to be counted as one vowel.

features are closely related to language models. They have been included in the approaches of Briscoe et al. (2010) and Yannakoudakis et al. (2011) to capture lexical and - in the case of higher order ngrams - structural information.

Language models draw their power form distributional probabilities and thus work best when a large amount of domain specific training data is available. This was not the case for the *MERLIN* data. Petersen and Ostendorf (2009) faced the same problem, when training readability models on the *Weekly Reader*, an educational magazine at different grade levels. As a solution they trained their language models on separate data sets (Petersen and Ostendorf, 2009). Feng (2010) who, like Petersen and Ostendorf (2009), worked with the *Weekly Reader* data, criticized this approach and proposed a leave-one-out approach for training language models directly on the *Weekly Reader* data.

We rejected the idea of Feng (2010)'s leave-one-out approach, because the amount of data offered by the *MERLIN* data would probably still be too small. Compared to the *Weekly Reader* data set used by Feng (2010), there are less documents - 1000 *MERLIN* essays in total compared to 1629 articles (Feng, 2010, 46) form the *Weekly Reader*. Additionally, the average number of words per document is much larger in the *Weekly Reader* (about 130-330 words) than in *MERLIN* (about 33-218 words). Another reason for using a separate data set for language model training is that the essays in *MERLIN* were written on three different tasks per exam level. It can be assumed that an essay shares more vocabulary with essays that were written on the same task than with those written on different tasks. This could cause the language model scores to represent the tasks rather than the proficiency levels.

Therefore separate training corpora, that we collected from the web, were used for language modeling. 2000 articles from *News4Kids* (`http://www.news4kids.de`), a German website which adapts news for children was used to represent easy texts. The *difficult* language model was trained on 2000 articles from the website of the German news channel *NTV* (`http://www.n-tv.de`). The same data has been used for building language models for German readability classification by Hancke et al. (2012).

Schwarm and Ostendorf (2005) and Feng (2010) suggested that mixed models, that combine words and parts-of-speech, are more effective for readability assessment than simple word based models. But Petersen and Ostendorf (2009) reached the opposite conclusion. Therefore both types of models were included.

The language models were prepared and evaluated as described in Hancke et al. (2012): All words were converted to lowercase and except for sentence

final punctuation, all punctuation was removed. As proposed by Petersen and Ostendorf (2009), a bag-of-words classifier was first trained with the language modeling data set (*News4Kids* and *NTV*). Information Gain (Yang and Pedersen, 1997) was used for feature selection. All words below an empirically determined threshold were replaced by their part of speech. *SRI Language Modeling Toolkit* (Stolcke, 2002) was used for training unigram, bigram and trigram models on the *words-only* and *mixed word/part of speech* corpora representing easy and difficult texts. For all models, Kneyser-Ney smoothing (Chen and Goodman, 1999) was selected as smoothing technique. This resulted in twelve language models.

The perplexity scores (Equation 3) from all twelve language models were included as features for proficiency assessment. Perplexity is an information-theoretic measure that indicates the fit between a text and a language model. To obtain these scores the spelling-corrected version of the *MERLIN* data was used. This should minimize cases where ngrams were not recognized due to spelling errors. All language model score features can be seen in Table 12.

$$PP = 2^{H(t|c)}, where H(t|c) = \frac{1}{m} log_2 P(t|c) \tag{3}$$

Level of Difficulty	Word Based Model	Mixed Model
Easy	Unigram perplexity	Unigram perplexity
	Bigram perplexity	Bigram perplexity
	Trigram perplexity	Trigram perplexity
Difficult	Unigram perplexity	Unigram perplexity
	Bigram perplexity	Bigram perplexity
	Trigram perplexity	Trigram perplexity

Table 12: The twelve perplexity scores used as language model features

5.3 Syntactic Features

Syntactic characteristics and measures have been deployed as features for proficiency assessment (Briscoe et al., 2010), but also in L2 developmental studies (Hawkins and Buttery, 2009, 2010; Lu, 2010; Biber et al., 2011) and readability assessment (Petersen and Ostendorf, 2009; Feng, 2010; Vajjala and Meurers, 2012). However, all these approaches worked with English texts. Hancke et al. (2012) adapted a number of syntactic features previously used in L2 developmental studies (Lu, 2010) and readability assessment (Petersen and Ostendorf, 2009; Feng, 2010; Vajjala and Meurers, 2012)

for readability assessment on German texts. We used these features as the foundation of our syntactic feature set and added several new indices.

5.3.1 Parse Rule Features

Briscoe et al. (2010) and Yannakoudakis et al. (2011) used the frequencies of parse rule names to gain information about the grammatical structures in learner texts. They extracted parse rule names from the parse trees produced by the *RASP* parser. While these types of features are not theoretically well informed, they allow a very comprehensive inclusion of the syntactic structures used by the learners. Looking at the distributions of those structures might point to indicative constellations.

We included similar features, based on the trees produced by the *Stanford Parser*. Parse tree rules were extract by the means of listing all local trees (Figure 7) for all parse trees. As a corpus for parse rule name extraction a subset of 700 articles form the *NTV* corpus was used. From the extracted parse rule names, a vector was formed. For each essay in the *MERLIN* corpus, the frequencies of the parse rule names in this vector were counted and divided by the number of words in the essay. Additionally experiments were made with just indicating whether a parse rule was present in an essay or not by putting the value to either zero or one.

```
(NUR PWAV $.)
(VP PP ADJD NP-OA VVPP)
(CNP-SB NE $, NN KON NN)
(PP APPR PRF)
(S KOUS NP-DA NP-OA VVFIN)
(VP PP NP-OA PP VVINF)
(PP APPR CNP PP)
(S CARD VVFIN NP-SB PP)
(NP ART NN PP $.)
(AP PP)
```

Figure 7: Examples for parse tree rules.

5.3.2 Dependency Features

Yannakoudakis et al. (2011) included the sum of the longest distances in word tokens between a head and a dependent in a grammatical relation as an indicator of syntactic sophistication. In readability assessment, the length of dependencies has also been used in various shades. Dell'Orletta et al. (2011) suggested the average length of dependencies in words as a

dependency-based alternative to phrase length. Additionally they examined the number of dependents per verb as an indicator of readability. Vor der Brück and Hartrumpf (2007); Vor der Brück et al. (2008) used the length of dependencies headed by specific parts-of-speech, such as number of words per NP and VP, as features for their readability classifier.

In this work, the following dependency based features were included: the maximum number of words between a head and a dependent in a text, the average number of words between a head and a dependent per sentence, the average number of dependents per verb (in words) including and excluding modifiers, and the number of dependents per NP (in words).

5.3.3 Parse Tree Complexity and Specific Syntactic Constructions.

As a starting point, we included the syntactic features that Hancke et al. (2012) adapted for German readability assessment from various sources. Features adapted from readability assessment comprised average parse tree height and the average length and number of NPs, VPs and PPs per sentence (Petersen and Ostendorf, 2009; Feng, 2010). To this set Hancke et al. (2012) added VZs ('zu'-marked infinitive phrases) per phrase as a language specific feature.

Syntactic Features from SLA	Formulas
Avg. Length of a Clause	# W / # C
Avg. Sentence Length	# W / # S
Avg. Length of a T-Unit	# W / # TU
Avg. Num. Clauses per Sentence	# C / # S
Avg. Num. T-Units per Sentence	# TU / # S
Avg. Num. Clauses per T-Unit	#C / # TU
Avg. Num. Complex-T-Units per T-Unit	# comp. TU / # TU
Avg. Num. Dep. Clause per Clause	# DC / # C
Avg. Num. Dep. Clause per T-Unit	# DC / # TU
Avg. Num. Co-ordinate Phrases per Clause	# CP / # C
Avg. Num. Co-ordinate Phrases per T-Unit	# CP / # TU
Avg. Num. Complex Nominals per Clause	# compl. Nom. / # C
Avg. Num. Complex Nominals per T-Unit	# compl. Nom. / # TU
Avg. Num. VPs per T-Unit	# VP / # TU

Table 13: Syntactic features from Hancke et al. (2012) based on features form SLA.

The measures from second language acquisition adapted by Hancke et al.

(2012)[27] aim at capturing a learner's syntactic development by examining specific syntactic properties. Complexity as reflected in the length of production units is measured by average length of sentences, clauses and T-Units (Lu, 2010; Vajjala and Meurers, 2012). Sentence complexity is reflected by the number of clauses per sentence. Ratios like dependent clauses per clause and number of complex T-Units per T-Unit are used to capture the amount of embedding, while the amount of coordination is represented by e.g., the number of coordinated phrases per clause and the number of T-Units per sentence. Finally, the complexity of particular structures is considered (e.g., complex nominals per clause).

Syntactic features from readability assessment	Formulas
Avg. Num. NPs per Sentence	#NP / # S
Avg. Num. VPs per Sentence	# VP / # S
Avg. Num. PPs per Sentence	# VZ / # S
Avg. Num. VZs per Sentence	# PP / # S
Avg. Num. NPs per Clause	# NP / # C
Avg. Num. VPs per Clause	# VP / # C
Avg. Num. PPs per Clause	# PP / # C
Avg. Num. VZs per Clause	# VZ / # C
Avg. Length of a NP	sum(len(NP)) / # NP
Avg. Length of a VP	sum(len(VP)) / # NP
Avg. Length of a PP	sum(len(PP)) / # NP
Avg. Num. Dep. Clauses per Sentence	# DC / # S
Avg. Num. Complex T-Units per Sentence	#compl. TU/ # S
Avg. Num. Co-ordinate Phrases per Sentence	# CP / # S
Avg. Parse Tree Height	sum(parseTreeHeight) / # S

Table 14: Syntactic Features from Hancke et al. (2012) based on previous work on readability assessment.

Inspired by different sources, we added several other syntactic features. The average number of non-terminal nodes per sentence, clause and word were included following Feng (2010). The internal complexity of NPs was measured by taking into account the number of modifiers per NP as was suggested by Graesser et al. (2004). Additionally the complexity of VPs was measured in the same way.

The number of passive voice constructions was counted per clause and by sentence. This feature was inspired by work on text simplification: Siddharthan (2002) identified passive constructions to transform them into ac-

[27]They follow Vajjala and Meurers (2012) who first deployed these measures for readability assessment

passive : `(wird|werden|wurden)/VAFIN .* */VVPP`
Futur 2 : `(wird|werden|wurden)/VAFIN .* */VVPP haben/VAINF`

Figure 8: Regular expressions used to identify passive voice

tive voice, because active voice is supposedly simpler. Grounded on this hypothesis it was interesting to investigate passive voice as a feature for proficiency assessment. In German, there is also the 'Zustandspassiv' (*Das Fenster ist geöffnet.*), which we did *not* included here.

To identify passive voice, we used dependency parsing and regular expressions (Figure 8). In passive voice, the inflected verb is a form of *werden* followed by a participle. The same construction is also part of Futur 2, where it is, however, followed by *haben*. For each verb in a sentence, all its verbal dependents were extracted from the dependency parse. Each verb was represented by its form and part of speech (`wordform/POS-Tag`). The head verb was concatenated into a string with its dependents. On this string, the regular expressions described above was applied to find passive voice.

A more detailed analysis of other specific constructions was inspired by Biber et al. (2011). In their article they challenged the common practice of relying on T-Unit and clausal subordination based measures for the assessment of grammatical complexity in writing. They investigated 28 grammatical features that were based on the grammatical types and grammatical functions of clauses and phrases (Biber et al., 2011).

In this work, it was not possible to include the full range of Biber et al. (2011)'s indicators of complexity. Biber et al. (2011) identified some features manually such as the syntactic functions of prepositional phrases. For other features, external resources would be needed, e.g., a list of nouns that control complement clauses[28] (Biber et al., 2011). Inspired by Biber et al. (2011)'s work, we included a more fine grained analysis of dependent clauses. Finite dependent clauses that start with a conjunction or pronoun, finite dependent clauses that do not start with a conjunction or pronoun, and non-finite dependent clauses ('satzwertige Infinitive') were set apart. Finite clauses that start with a conjunction or pronoun were further differentiated into interrogative, conjunctional and relative clauses.

[28] *GermaNet* verb frames, that contain subcategorization information for the verbs could be used to enable a more fine grained analyzes. However an implementation was not possible within the bounds of this work.

Other Syntactic Features	Formulas
Avg. Num. Non-Terminals Per Sentence	# NTs / # S
Avg. Num. Non-Terminal Per Words	# NTs / # W
Avg. Num. Modifiers Per NP	# modifiersInNPs / # NPs
Avg. Num. Modifiers Per VP	# modifiersInVPs / # VPs
Passive Voice - Sentence Ratio	# passiveVoice / # S
Passive Voice - Clause Ratio	# passiveVoice /# C
Dep. Clauses with Conj. to dep. Clause Ratio	# DC w. Conj. / # DC
Conjunctional Clauses Ratio	# Conj. C / # dep. C w. Conj.
Interrogative Clauses Ratio	# Inter. C / # dep. C w. Conj.
Relative Clauses Ratio	# Rel. C / # DC w. Conj.
Dep. Clauses w.o. Conj. to dep. Clause Ratio	# DC w.o. Conj. / # DC
'satzwertige Infinitive' to Clause Ratio	# satzInf / # DC

Table 15: Other syntactic features that measure parse tree complexity and examine specific constructions.

5.4 Morphological Features

Morphological indicators have proven to be valuable features for proficiency classification for Estonian (Vajjala and Loo, 2013) and for readability assessment for various languages with a rich morphology (Dell'Orletta et al., 2011; Francois and Fairon, 2012; Vor der Brück and Hartrumpf, 2007; Vor der Brück et al., 2008; Hancke et al., 2012).

German,too, has a rich inflectional and derivational morphology. Inflectional morphemes convey a range of grammatical meanings. German nominal declension has four cases and several different declension paradigms. Different verb forms and inflectional morphemes of the finite verb express person and number (e.g., *ich gehe* [*I go*], *du gehst* [*you go*]) as well as tense and mood.

Compounding is very productive in German word formation. Words with different parts-of-speech can be combined (e.g., *Mauseloch* [*mouse hole*], *Schwimmbad* [*swimming pool*], *Grosseltern* [*grandparents*]). Prefixation (*Gebäude* [*Building*]) and suffixation are also very common and diverse. For example, there is nominalization with overt suffixes (*regieren*[*govern*] – *Regierung* [*government*]) or without an overt suffix (*laufen* [*to run*] – *der Lauf*[*the run*]) (Hancke et al., 2012).

In this work, the morphological features proposed for readability assessment by Hancke et al. (2012) were taken as a vantage point for exploring the impact of morphological features on proficiency assessment. Additionally, tense was explored in more detail, using automatically extracted tense patterns.

5.4.1 Inflectional Morphology of the Verb and Noun

Following Hancke et al. (2012) we extracted a broad range of features based on verbal inflection including person, mood and type of verb (finite, non-finite, auxiliary). Additionally we included nominal case information as features. All inflectional features (Tables 16 and 17) were automatically extracted from the output of the *RFTagger*.

Verb Features	Formulas
Infinitive-Verb Ratio	# infinitive Vs / # Vs
Participle-Verb Ratio	# participle Vs / # Vs
Imperative-Verb Ratio	# imperative Vs / # Vs
First Person-Verb Ratio	# 1st person Vs / # finite Vs
Second Person-Verb Ratio	# 2nd person Vs / # finite Vs
Third Person-Verb Ratio	# 3rd person Vs / # finite Vs
Subjunctive-Verb Ratio	# subjunctive Vs / # finite Vs
Finite Verb-Verb Ratio	# finite Vs / # Vs
Modal-Verb Ratio	# modal Vs / # Vs
Auxiliary-Verb Ratio	# auxiliary Vs / # Vs
Avg num. Verbs per Sentence	# Vs / # S

Table 16: The features based on the inflectional morphology of the verb

Noun Features	Formulas
Accusative-Noun Ratio	# accusative Ns / # Ns
Dative-Noun Ratio	# dative. Ns / # Ns
Genetive-Noun Ratio	# genitive Ns / # Ns
Nominative-Noun Ratio	# nominative Ns / # Ns

Table 17: The features based on inflectional morphology of the noun

5.4.2 Derivational Morphology of the Noun

Examining nominal suffixes is not only interesting because derived words are supposedly more complex than simple words (Vor der Brück et al., 2008). Word stems of Germanic origin can often be combined with other suffixes than word stems that have Greek or Latin roots (e.g., *Lingu**ist*** vs. *Sprachwissenschaft**ler***).

To capture the information encoded in nominal suffixes, we counted the occurrence of each suffix in a list (Table 18) originally compiled by Hancke

et al. (2012). The list includes all different gender and number forms for each suffix. The counts for all forms of a suffix were accumulated in order to not get different counts for the plural and singular forms or for different genders of the same suffix. Only polysyllabic words were considered in order to exclude simple nouns that are homomorphs of a derivational morpheme (e.g., *Ei* [*egg*] vs. suffix -*ei*).

Suffix	Further Suffix Forms
ant	anten, antin, antinnen
arium	arien
ast	asten, astin, astinnen
at	ate
ator	atoren, atorin, atorinnen
atur	aturen
ei	eien
er	erin, erinnen
ent	ents
enz	enzen
eur	eure, eurin, eurinnen
heit	heiten
ist	isten, istin, istinnen
ion	ionen
ismus	ismen
ität	itäten
keit	keiten
ling	lingen
nis	nisse
schaft	schaften
tum	tümer
ung	ungen
ur	
werk	werke
wesen	

Table 18: List of German derivational suffixes used

Hancke et al. (2012)'s experiments with different ratios showed that the best results were obtained by dividing the suffix counts by the number of tokens in an essay. Therefore the suffix features were calculated by counting the frequencies of all suffixes in an essay and then dividing these counts by the number of all tokens in the essay. Additionally the ratio of all derived nouns to all nouns was included as a feature.

5.4.3 Nominal Compounds

Following Hancke et al. (2012) the ratio of compound nouns to all nouns and the average number of words in a compound were included as features. Compounds were identified and split into their components with *JWordSplitter* 3.4[29]. It is unclear whether the use of compounds is indicative of a learner's linguistic proficiency. However, we think it is worth investigating. Especially for learners with native languages that do not use this word formation mechanism, compounds may be a stumbling block.

[29]http://www.danielnaber.de/jwordsplitter

5.4.4 Tense

To capture the usage of tense, 'tense patters' (Figure 9) based on the morphologically rich tags of the *RFTagger* were extracted. This approach was preferred to using regular expression for matching pre-defined tenses because we believe that it allows a better coverage of possible constellations. The tense patterns could be very interesting for a linguistic characterization of the CEFR levels, because they might reveal what tenses the learners know at each level.

John fragt den Mann,
`VFIN.Full.Pres`
der neben ihm gewohnt hat,
`VFIN.Haben.Pres VPP.Full.Psp`
ob er die Katze füttern kann.
`VFIN.Mod.Pres VINF.Full.-`

Figure 9: Example for tense patterns

The patterns were extracted from the same subset of the *NTV* corpus as the parse rules (section 5.3). From the dependency parse, all those verbal children of all verbs were collected, where the relation did not cross the boundaries of a subordinated clause (tags `KOUS`, `KOUI`) or a conjunction (tags `KON`). This restriction was necessary because subordinated clauses are often the dependent of the finite verb in the main clause. Relative clauses were not considered boundaries, since they do not attach to verbs. The verb's children found in this way were substituted by their *RFTagger* tags. The information about number, person and mood was removed, so that only the tense information remained.

To make sure that only the longest pattern was included, the extraction routine checked whether the previous match was included in the current match. If so, the current pattern replaced the previous pattern instead of being added as a new pattern. For each script, the frequency of each pattern was counted and divided by the number of tokens in the script. Alternatively, we experimented with using a binary scale, where one indicates that a pattern was present in an essay, and zero indicates that it was not present.

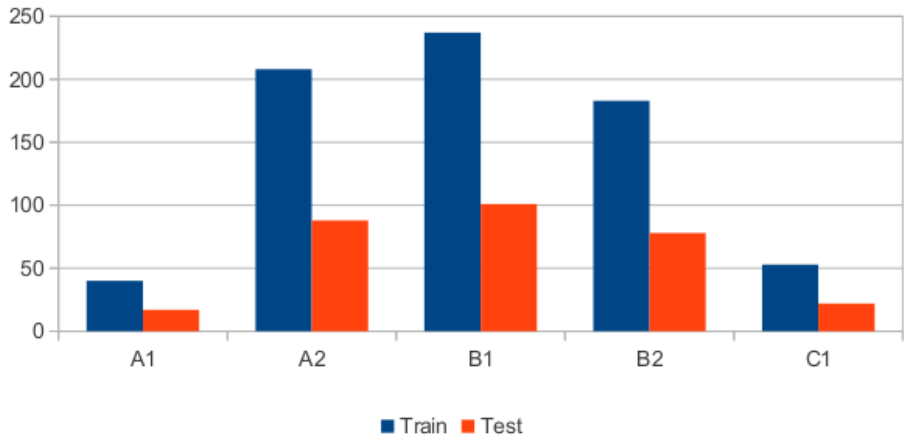

Figure 10: Number of samples per class in the training and test set

6 Experiments and Results

6.1 Experimental Setup

All experiments were conducted on the *MERLIN* data set described in section 3. The data was preprocessed as described in Section 4.1. For supplementing statistical analysis we used R. The *WEKA* machine learning toolkit (version 3.7.6) (Hall et al., 2009) was used for all experiments involving machine learning. All feature values were normalized with the *WEKA* filter *Normalize*, such that all values fall into the interval [0,1]. Most experiments were performed with 10-fold cross validation on the whole data set, as well as on a separate training and test set. For creating the training and test set, 2/3 randomly chosen samples from each class were added to the training set and 1/3 to the test set.

Splitting the data set regarding no other factor than class (= essay rating level) resulted in an uneven distribution of exam types across the classes and the training and test set (Figure 11). While in theory this should not have any influence on the results, in practice it is quite likely that the essay rating level (=class) is not independent of the exam type. The correlation between exam type and essay rating level is 0.8 (Pearson's correlation).

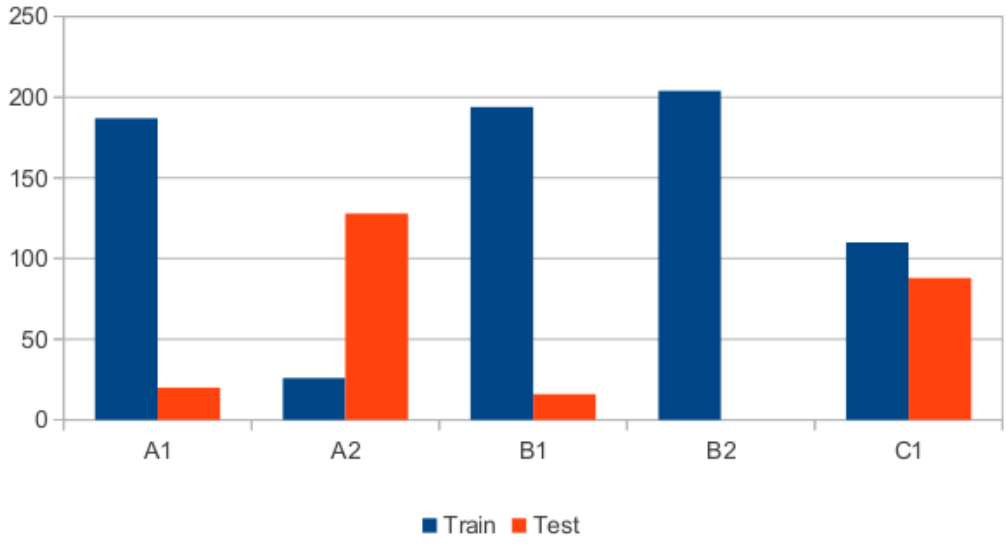

Figure 11: Exam type distribution across the classes in the training and test set.

6.2 Comparison of different Algorithms for Classification

As a first step we experimented with different machine learning algorithms. We tested SMO, Naive Bayes and J48 Decision Tree. Naive Bayes and J48 were used with their standard configurations[30] in *WEKA*, except that for SMO, the normalization of the input was turned off since the data was already normalized. All features were used for building the models.

The results can be compared to a majority baseline, which represents the performance if every sample would be classified as the class with the most samples. For the sake of comparison we also built a regression model with *WEKA*'s Linear Regression (*LinearRegression*[31]) using cross validation for training and testing. It resulted in a good correlation (r =0.78) but also in a relatively high root mean squared error (0.68).

Comparing the results of the three different classification algorithms showed that SMO performed better than the other classifiers. Therefore it was chosen for all further experiments. This result was expected since Support Vec-

[30]Naive Bayes: weka.classifiers.bayes.NaiveBayes ; J48: weka.classifiers.trees.J48 -C 0.25 -M 2 ; SMO: weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 2 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"

[31]Automatic feature selection and the elimination of redundant features was turned off in the configuration

	Holdout		CV on all data	
Classifier	Accuracy	F-Measure	Accuracy	F-Measure
Majority Baseline	33.0	16.3	33.0	16.3
SMO	57.2	56.4	64.5	64.0
Naive Bayes	41.17	40.3	48.2	49.3
J48	48.7	47.8	56.5	56.2

Table 19: Comparison of different classifier algorithms.

tor Machines are especially suitable for numeric features and larger feature spaces. They have been previously used for similar problems with good results (Petersen and Ostendorf (2009); Vajjala and Meurers (2012)).

It is remarkable that irrespective of the classifier, the cross validation results were persistently about 7 % better than the results with the separate training and test set. There are several possible explanations. The models built with cross validation were probably more accurate because more training data was available. The models' variance could be high, which would mean that they probably over-fit the training data and therefore do not generalize well on different test sets. The unequal distribution of exam types across training and test set could be another reason. We examined the SMO model in more detail by looking at the performance of each cross validation fold separately. The highest accuracy was 10.7 % better than the worst (59.2 %). The worst result for a fold in cross validation, however, was still 2 % better than the accuracy for our holdout test set. This indicates that the relatively high variance of the model is responsible for the difference in results between cross validation and holdout estimation in a greater degree than the amount of training data.

6.3 SMO Configuration and Optimization

For all further experiments we used SMO with a polynomial kernel of exponent 1. We also experimented with other exponents and with RBF kernel, but the polynomial kernel with exponent 1 exceeded the other options. The polynomial kernel implements a function $(xy)^n$, which computes the dot product of the vectors x and y and raises it to the power of n. Choosing $n = 1$ in fact results in a linear model (Witten and Frank, 2005).

One of the disadvantages of Support Vector Machines is their sensitivity to the tuning of certain parameters. When using the polynomial kernel, once an exponent is chosen, the cost parameter C should be optimized in order to get robust results (Harrington, 2012). The cost parameter manages the trade

off between making the margin as large as possible and keeping a distance of at least one between the samples (Harrington, 2012). If C is too large the model will tend to over-fit. If it is too small it tends to under-fit.

In order to optimize the cost parameter in our experiments we used *WEKA*'s meta classifier *CVParameterSelection*. It performs 10 fold cross validation on the training data to find the optimal parameter value in a specified range. We chose a range from 1 to 10 in steps of 1. One of the main advantages of using the meta classifier is that it allows the use of 10 fold cross validation for parameter selection and for model building / performance evaluation. It performs an "inner" cross validation for parameter selection on each "outer" cross validation for model building / performance evaluation. The disadvantage is, that this procedure is computationally expensive and time consuming.

6.4 Measuring the Contribution of Different Feature Types

Syntactic Features	r
Avg. Num. Dep. Clause per Clause	0.58
Avg. Length of a T-Unit	0.57
Avg. Num. Dep. Clause per T-Unit	0.56
Avg. Num. VZs per Sentence	0.55
Avg. Num. Complex-T-Units per T-Unit	0.54
Avg. Num. VPs per T-Unit	0.54
Avg. Num. Dep. Clauses per Sentence	0.53
Avg. Num. VZs per Clause	0.52
Dep. Clauses with Conj. to dep. Clause Ratio	0.50
Avg. Num. Clauses per T-Unit	0.50

Table 20: The ten syntactic features that correlate best with proficiency level.

To enhance our understanding of the feature groups we calculated the correlation of each feature with proficiency level (= essay rating level), based on the training set. The 10 features of each group that correlated best with proficiency level are shown in Tables 20 - 23. Although the parse rule (PR) and tense features (TEN) conceptually belong to the syntactic and morphological feature group respectively, they were treated as separate groups because they differ from the other groups in several respects. While the other features are mostly theory driven, the tense and parse rule features are data driven. Additionally they contain a far larger amount of features than the other groups, and there is more sparsity. Features like average sentence

Morphological Features	r
Derived Nouns-Nouns Ratio	0.59
Nominative-Noun Ratio	-0.54
Second Person-Verb Ratio	-0.46
Avg. Compound Depth	0.45
Dative-Noun Ratio	0.40
Avg. Num. Verbs Per Sentence	0.40
Infinitive-Verb Ratio	0.38
keit	0.37
ion	0.35
Third Person-Verb Ratio	0.35

Table 21: The ten morphological features that correlate best with proficiency level.

length and Type-Token Ratio can be reliably calculated for every text while many tense constellations and parse rules occur only in some texts.

Lexical Features	r	Language Model Features	r
Text Length	0.81	Unigram Plain Difficult	-0.47
Root Type-Token Ratio	0.78	Unigram Plain Easy	-0.46
Corrected Type-Token Ratio	0.78	Bigram Mixed Difficult	0.42
Corrected Verb Variation 1	0.75	Trigram Mixed Easy	0.34
Squared Verb Variation 1	0.74	Bigram Mixed Easy	0.33
Type-Token Ratio	-0.57	Bigram Plain Difficult	0.32
HDD	0.54	Trigram Plain Easy	0.25
Google Spell Check Error Rate	-0.47	Bigram Plain Easy	0.15
Avg. Num. Characters per Word	0.47	Unigram Mixed Difficult	0.09
Avg. Num. Syllables per Word	0.44	Trigram Mixed Difficult	0.07

Table 22: Ten lexical and language model features that correlate best with proficiency level.

The lexical group contains 5 features that correlate better than the best correlations in all other groups ($r > 0.7$). The other 5 features among the 10 best of this group still have good correlation values. Notably Type-Token Ratio has a negative correlation with proficiency level, while its variants show positive correlations. This may be explained with Type-Token Ratio's dependency on text length. For extremely short texts it is easier to obtain a high Type-Token Ratio than for longer texts. This negative correlation might indicate that the increase in text length in the higher levels is greater than the increase in the richness of vocabulary. Text length and Type-Token Ratio have a correlation of -0.65. Notably the correlation of other features with

text length are much higher: Root Type Token Ratio (r = 0.86), Squared Verb Variation (r = 0.81) and Corrected Verb Variation (r = 0.84). This might indicate that they are not actually good measures for proficiency but encode text length instead.

In the syntactic group none of the features correlate as highly with proficiency level as the 5 best features in the lexical group. However, all ten values are above 0.5. More than half of the top 10 correlating syntactic features encode embedding. The features of the morphological group all in all correlate slightly worse than those in the syntactic group. Most notable is the relatively high (r = 0.59) correlation of Derived Nouns-Noun Ratio which encodes the overall use of nominal derivation and nominalization. Nominative-Noun Ratio has a negative correlation and could either encode the use of conceptually simple sentences or grammar mistakes - another case might have been correct in some places where nominative appeared. A detailed study would be interesting. Second Person-Verb Ratio might be seen as a sign of personal and non-formal style. It is to be suspected that its use depends on the task, the essays were written for. The fact that it correlates better with exam type (r =-0.58) than with proficiency level supports this claim.

From the parse rule group ROOT_NUR had a high negative correlation with proficiency level. NUR means "Non Unary Root". It is an artifact used by the *Stanford Parser* when converting trees from *NEGRA* to Penn Treebank format trees. It is introduced if a tree has multiple children under the ROOT node (personal correspondence with Christopher Manning). It is hard to interpret this in the context of proficiency levels, but the negative correlation and its occurrence in non standard trees indicate that it reflects word order mistakes or missing punctuation. The parse rule feature with the second highest correlation encodes 'satzwertige Infinitive' like *Computer* ***zu spielen***. It should be mentioned that most of the over 3000 parse rule features exhibited a correlation of lower than 0.1.

In the language model group the easy and difficult Plain Text Unigram Perplexity scores were the most indicative features. It is however intriguing that both correlations are negative. It would have been expected to see a negative correlation for the model that represents easy language and a positive correlation for the model that represents difficult language. In the tense group all correlations were quite low. The best values were achieved by an infinite modal followed by an infinite verb as in *Sie können gehen. [You may go.]* and by infinitives with 'zu'.

To further examine the predictive power of each group of indicators we built a classifier with each feature group individually. Text length was chosen as an additional baseline due to its high correlation with proficiency level. The results showed that all feature groups outperformed the majority base-

Parse Rule Features	r	Tense Feature	r
ROOT_NUR	-0.62	VINF.Mod.- VINF.Full.-	0.21
VZ_PTKZU_VVINF	0.49	VINF.Full.zu	0.21
S_NP-SB_VVFIN_PRF_NP-SB_$.	-0.40	VFIN.Aux.Pres VPP.Full.Psp	0.18
NP-SB_ADJA_ADJA_NN_PP	-0.39	VFIN.Full.Pres	-0.17
NP-SB_NN_NE	-0.32	VINF.Aux.- VPP.Full.Psp	0.15
NP-OA_PPER	-0.31	VFIN.Mod.Pres.VFIN.Sein.Pres	0.14
VP_VVPP_$,_CVP	0.31	VINF.Full.- VFIN.Full.Past	0.14
VP_NP-DA_NP-OA_VVIZU	0.31	VFIN.Mod.Pres.VINF.Sein.-	0.13
NP-OA_NE	-0.28	VFIN.Mod.Pres	-0.12
ROOT_NP	-0.28	VIMP.Full.Sg	-0.12

Table 23: Ten parse rule and tense features that correlate best with proficiency level.

line, but only the lexical feature group performed better than the text length baseline in terms of accuracy in the cross validation scenario. The lexical and morphological group performed better than text length when comparing F-measure. In the holdout scenario no feature outperformed the baseline of text length in terms of accuracy, and only the lexical group exceeded it in terms of F-measure. With both methods of evaluation the lexical and morphological groups performed best, and parse rule and tense features yielded the poorest results. The results were mostly congruent with the insights gathered from looking at the ten best correlating features from each group. The lexical features performed best. Although displaying slightly lower correlations with proficiency level among the top 10 features, the classifier built with the morphological features exceeded the classifier built with the syntactic features. As expected the parse rule, language model and tense groups performed worst.

Further investigating the parse rule and tense feature groups however, we found out that representing those features as binary instead of frequency weighted vectors immensely improved their performance (7 % and 18 % respectively). Looking at the values of the tense features before normalization revealed that they were mostly 0 and even the non zero values were mostly close to 0. The maximum value was 2. For the parse rule features the situation was similar, with the maximum feature value being 0.17. This could explain why they yield better result when used as binary features: if the feature values are mostly very close to zero they may not be separable by the support vector machine, especially when mixed with larger values.

We conclude that it would be optimal to use the tense and parse rule features as binary features. However, combining binary and numeric features

		Holdout		CV on all data	
Name	#	Accuracy	F-Measure	Accuracy	F-Measure
Majority Baseline	-	33.0	16.3	33.0	16.3
Text Length (BL)	1	61.4	55.7	63.9	59.4
SYN	47	53.6	49.9	58.2	54.5
PR	3445	49.0	47.2	57.1	56.0
LEX	46	60.5	58.9	68.7	67.4
LM	12	50.0	46.7	54.6	51.0
MORPH	41	56.82	55.7	61.8	60.1
TEN	230	38.5	36.8	44.2	41.7

Table 24: Performance of each group of features individually.

Name	Accuracy	F-Measure
PR	61.1	60.5
TEN	56.8	54.7

Table 25: Performance of the tense and parse rule groups as binary features. Training and testing with 10-fold CV on all data.

into one model would require more sophisticated machine learning techniques and is not manageable in the scope of this work. It is still an interesting fact that might be picked up in the future.

6.5 Combining Different Feature Groups

After having examined the performance of each feature group individually we tested the performance of all possible combinations of feature groups. This experiment might offer some insights on which of the feature groups complement each other. For each two, three and four class combinations, the three most successful combinations are reported in addition to the combination of all feature groups. The experiment was conducted with the separate training and test as well as with cross validation on the whole data set.

In comparison to using the groups individually, the performance mostly improved when combining two or three feature groups . However, for four groups and when using all features, the performance of the classifiers dropped considerably below the performance of the best individual group. This suggests that too much useless features accumulated, which can have a negative effect on the models' predictive power.

The best combination of feature groups with cross validation as evaluation was LEX_LM_MORPH. When using holdout estimation, the LEX_MORPH

CV on all data		
Name	Accuracy	F-Measure
SYN_LEX	69.7	67.3
LEX_MORPH	69.4	69.1
LEX_LM	69.1	68.0
LEX_LM_MORPH	70.1	69.9
SYN_LEX_LM	69.9	66.9
SYN_LEX_MORPH	68.5	67.9
SYN_LEX_LM_MORPH	68.9	68.6
LEX_LM_MORPH_TEN	68.8	68.3
SYN_LEX_LM_TEN	65.5	64.0
ALL	64.5	64.0

Table 26: Three best performing two, three and four class combinations and the performance of all features (CV).

Holdout		
Name	Accuracy	F-Measure
LEX_MORPH	61.1	60.8
LEX_TEN	59.8	59.3
LEX_LM	59.4	59.0
LEX_LM_MORPH	61.1	60.58
SYN_LEX_MORPH	58.5	58.0
LEX_LM_TEN	57.8	57.6
SYN_LEX_LM_MORPH	58.8	58.4
SYN_LEX_LM_PR	57.8	57.3
LEX_LM_MORPH_TEN	57.8	57.2
ALL	57.2	56.4

Table 27: Three best performing two, three and four class combinations and the performance of all features (holdout).

and LEX_LM_MORPH were equal in accuracy. It is interesting to see that the different methods for model training and testing produced different rankings. Investigating the reasons for these discrepancies would be very interesting, but has to be deferred to future work.

In addition to the classification experiments, we examined the intercorrelations of individual features, mainly to detect high intercorrelations across groups. Notably, Average Number of Verbs per Sentence from the morphological group correlated highly with several syntactic features, for instance Average Sentence Length (r = 0.91) and Average Longest Depen-

dency per Sentence (r = 0.86). Furthermore there was a high correlation between Participle-Verb Ratio and Passive Voice per Clause (r = 0.86). Unsurprisingly there were high intercorrelations between some elements of the syntactic and parse rule group, for example Average VZ Frequency and the corresponding parser tag VZ PTKZU VVINF (r = 0.89). Many high intercorrelations also occurred between features of the tense and the parse rule group.

From the declining results for combinations of more than three groups it can be concluded that these combinations have potential but that too much noise disturbs the SMO. Feature selection can be applied to find a good combination of features. For future work it might be interesting to train an ensemble classifier instead of combining feature groups.

6.6 Feature Selection

As the previous section has illustrated, using all available features together in one model does not result in the best classifier. Irrelevant features have a negative impact on most machine learning algorithms, slow them down and make the resulting models harder to interpret. It is therefore a common practice to perform attribute selection before classification (Witten and Frank, 2005). We used the *WEKA* implementation of correlation based feature selection (*CfsSubsetEval*). This method is independent of the machine learning algorithm used. It evaluates the merit of each feature in terms of correlation with the class but also takes redundancy among the features into account. Features that correlate highest with the class but have a low intercorrelation are preferred (Witten and Frank, 2005).

		CV on Training Set		Holdout	
Name	#	Accuracy	F-Measure	Accuracy	F-Measure
LEX_LM_MORPH	30	71.2	70.7	61.7	61.3
SYN_LEX_LM_MORPH	34	72.5	72.4	62.7	62.2
ALL	88	71.6	71.2	61.8	60.7

Table 28: Results after Feature Selection with CfsSubset Evaluation

We applied feature selection to the most successful combinations of three and four feature groups and to the whole feature set. Since *CfsSubsetEval* uses the whole training set for feature selection we conducted the following experiment on the separate training and test set only. Since the test set should not be used for judging the success of feature selection, the results are shown for 10 fold cross validation on the training set. Choosing a feature

set should based on these results only. However, since this is experimental work, we also show the results on the heldout test set to be able to compare their performance to their non feature-selected counterparts.

Interpretation	Features
Length / sophistication of production units	Avg. Sentence Length, Avg. Length of a T-Unit
Embedding	Dep. Clauses with Conj. to dep. Clause Ratio, Avg. Num. Non-Terminal Per Words
Verb phrase complexity	Avg. Num. VZs per Sentence, Avg. Length of a VP
Coordination	Avg. Num. Co-ordinate Phrases per Sentence
Use of passive voice	Passive Voice - Sentence Ratio
Script length	Text Length
Lexical richness	Type-Token Ratio, Root Type-Token Ratio, Corrected Type-Token Ratio, HDD, MTLD
Lexical richness w. respect to verbs	Squared Verb Variation 1, Corrected Verb Variation 1
Nominal style	Noun Token Ratio
Word length / difficulty	Avg. Num. Syllables per Word, Avg. Num. Characters per Word
Frequency of the words used in the script / Vocabulary Ease	Annotated Type Ratio, Ratio of Words In Log Frequency Band Two, Ratio of Words In Log Frequency Band Four, Unigram Plain Easy
Spelling errors	Ratio of Lex. Types not in Dlex, Google Spell Check Error Rate
Nominalization, use of derivational suffixes, use of words with Germanic stems	keit, ung, werk, Derived Nouns To Nouns Ratio
Nominal case	Genetive-Noun Ratio, Nominative-Noun Ratio
Verbal mood and person	Subjunctive-Verb Ratio , Second Person-Verb Ratio , Third Person-Verb Ratio

Table 29: The 34 features constituting the best performing model after feature selection.

The results showed that feature selection improved all three models. Generally it can be observed that with feature selection the performance of the

models became more similar. Their accuracies varied less than 1 %, while the difference between the best and the worst of the models before feature selection was 3.9 %. This suggests that feature selection indeed made the models more robust. With feature selection SYN_LEX_LM_MORPH (*Best34*) is the best performing model instead of LEX_LM_MORPH (*Best30*). The second best model consists of the 88 features selected from the whole dataset (*Best88*).

Table 29 shows all features in *Best34*. They were summarized into groups according to their possible interpretation. It can be observed that although *CfsSubsetEval* prefers features with a low intercorrelation, it still keeps intercorrelating features to a certain extent if their predictive power is convincing. Selected features with a highly intercorrelation were for example Squared Verb Variation and Corrected Verb Variation (r = 0.97) or Average Number of Characters per Word and Average Number of Syllables per Word (r = 0.94).

Best34 comprises 7 syntactic, 16 lexical, 1 language model and 9 morphological features. It is remarkable that most linguistic aspects, that were encoded in the model previous to feature selection, are represented in *Best34*. It has already been mentioned that text length has a high correlation with proficiency level. Consequently it was among the selected features. Vocabulary difficulty and lexical richness also seemed to be particularly indicative. They are represented by shallow features like the Average Number of Characters per Word but also by the general frequency of the vocabulary used in the essays as measured by *dlexDB* scores and by the Easy Plain Unigram language model. Additionally several Type-Token Ratio variations, that measure lexical richness, were selected, as well as lexical variation measures and Noun-Token Ratio. Notably also both features that measure spelling errors were among the best 34 features.

On the syntactic level shallow indicators such as Average Sentence Length were included as well as parse tree based measures of embedding and coordination, and the use of passive voice. From the morphological group, features that encode verbal mood and person, nominal declension and nominal suffixes were added. Finally it can be seen that none of the lexical relatedness measures or morphological noun compounding features were present in *Best34*, nor in fact *Best30* or *Best88*. It can be concluded that they are not particularly predictive of proficiency level.

To check if the model had become more stable after feature selection, we trained and tested a model based on *Best34* with cross validation on the whole data set, and examined the accuracies of each fold separately. Compared to the 10.7 % difference between the highest and lowest accuracy for a fold in the initial model (see section 6.2), the different between the

highest and lowest result was reduced to 8.56 %. This indicates that *Best34* is more robust.

The experiments showed that feature selection with *CfsSubsetEval* improved the classification results and lead to a feature set that encodes proficiency at different, conceptually satisfactory levels. Additionally the model became more robust. Therefore we conclude that applying feature selection with *CfsSubsetEval* for model optimization is a recommendable step.

6.7 Pass or Fail Classification for Each Exam Type Separately

In a real world scenario users might not want to consider all essays regardless of their exam type. They might want to know which people passed or failed a particular exam. Imagine a group of students had just finished an A2 exam. A student passes the exam if the essay is assigned at least the rating level that equals the exam type. As an example, if a student took an A2 exam, and was graded as A2 or above, she would have passed the exam. If she was graded as A1 she would have failed. While a user could still successfully use the classifier that we built in the previous sections to obtain the CEFR levels, and then decide who passed and failed, we also examined binary "pass or fail" classification for each exam type separately. The performance might be better as the classifier can be attuned specifically to a pass of fail scenario.

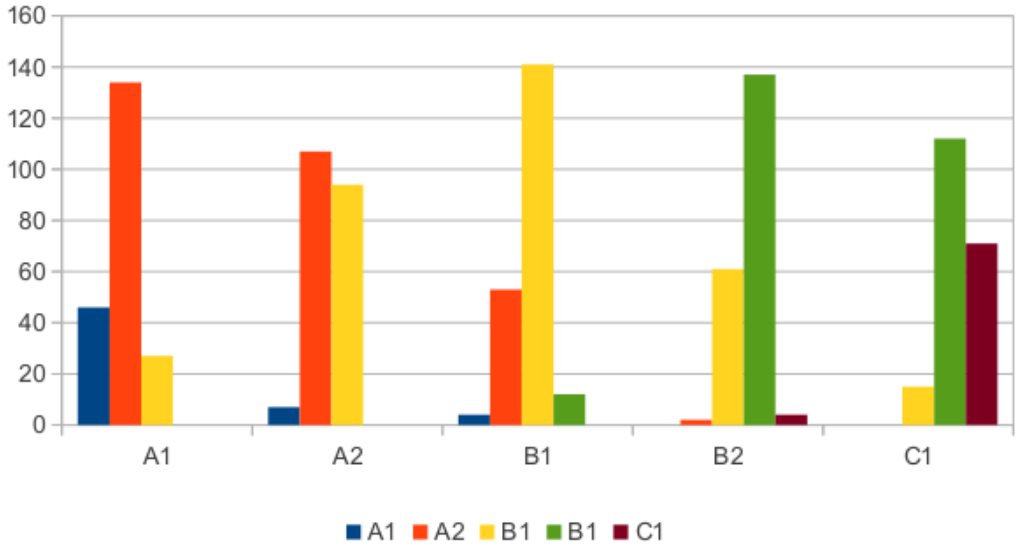

Figure 12: Distribution of classes for each exam level.

Figure 12 shows, that the class (essay rating level) distributions over each individual exam type are very unequal. Distributions, that are enormously skewed and offer very little samples for some classes, have an negative impact on machine learning algorithms. As most algorithms optimize the overall cost, for a distribution like *7 fail - 200 pass* most algorithms would almost certainly learn a majority classifier (classify all examples as pass). *Cost sensitive learning* can be used to compensate an uneven distribution, or to take into account that misclassifying one class can be more expensive or problematic than the other. Usually all types of errors have the same cost. Giving different weights to instances of different classes can be used to make some classification errors more expensive than others (Harrington, 2012; Witten and Frank, 2005). While adjusting the weights is relatively straightforward for a two class problem, it is much more complicated when multiple classes are involved. Also, it has been noticed that the same methods, that work well for two class problems, do often not perform as expected for multiclass problems (Abe et al., 2004). For more detailed information on cost sensitive classification in general see Ling and Sheng (2010). For a discussion of applying it to multiclass problems see Abe et al. (2004).

For cost sensitive classification we used *WEKA*'s meta classifier *CostSensitiveClassifier* with SMO as a base classifier and the option for re-weighting training instances according to the cost matrix. This implementation uses bagging to make the base classifier cost sensitive (Witten and Frank, 2005, 319-320) and is based on the approach described in Domingos (1999). All experiments were conducted with the *Best34* feature set. Because there were only about 200 samples per exam type, we chose leave-one-out for building and evaluating the classifiers. *CVParameterSelection* with 10 folds was used for selecting SMO's C parameter. In addition to accuracy, we report precision and recall per class.

The numbers in Table 30 show the results of experimenting with cost sensitive learning. As the exact costs are not known to us, we aimed at balancing the uneven distributions. The weights that we chose were guided by the class distributions. In a real world scenario statistics about the fail rate of an exam or expert knowledge might be used instead. It can be seen that the accuracies, and the precision values for the minority classes dropped when the costs were adjusted, but the precision values for the majority classes increased. Most importantly, the recall values for the minority classes improved. The experiment with A2 (Table 31) shows a classifier that was not designed to merely balance the distribution, but to prioritize the detection of "fail" cases. Therefore the cost for misclassifying "fail" as "pass" was set to twice the number of instances in the majority class.

We conclude, that for two class problems, cost sensitive learning provides

Name	Accuracy	Precision	Recall	Cost-Matrix
B1 fail	82.8	75.6	54.4	
B1 pass	82.8	84.6	93.5	
B1-c fail	74.3	51.9	73.7	0 3
B1-c pass	74.3	88.4	74.5	1 0
B2 fail	77.9	68.8	52.4	
B2 pass	77.9	80.8	89.4	
B2-c fail	73.5	55.2	76.2	0 153
B2-c pass	73.5	87.2	72.3	57 0
C1 fail	68.7	73.7	79.5	
C1 pass	68.7	57.4	49.3	
C1-c fail	67.7	77.9	69.3	0 71
C1-c pass	67.7	54.1	67.7	127 0

Table 30: Balancing skewed distributions with cost sensitive learning.

Name	Accuracy	Precision	Recall	Cost-Matrix
A2 fail	96.6	0	0	
A2 pass	96.6	96.6	1	
A2-c fail	85.1	14.7	71.4	0 400
A2-c pass	85.1	98.9	85.6	1 0

Table 31: Training a classifier that gives a higher priority to detecting "fail" cases.

an interesting opportunity to balance skewed data sets and influence the classification in favour of a particular class. How exactly this device should be used in automatic language proficiency assessment depends on the situation and has to be decided individually.

6.8 Classification with Rasch Corrected Scores

In addition to the actual ratings assigned by human annotators, the *MERLIN* project also offered a corrected version of the ratings. Multifaceted RASCH analysis was used to model effects that are associated with the properties of individual raters and generate the corrected scores (Bärenfänger, 2012). Classification is expected to work better with the RASCH corrected scores, since they are supposed to be more consistent (Briscoe et al., 2010). For comparison's sake we trained a classifier with all features on the whole data set using 10 fold cross validation. With an accuracy of 67.9 % and F-measure of 66.7 % the classifier built on the corrected data indeed performed better than the model previously built under the same conditions on the actual

human ratings (64.5 % accuracy, 64.0 % F-Measure). This result indicates, that some classification errors in our approach can be attributed to the inconsistencies of the human ratings. A more detailed analyses, that would include the manual inspection of some essays, could be enlightening.

6.9 Predicting the Exam Type

In this experiment we built a classifier for predicting the exam type instead of the essay rating level. There are several reasons why this is interesting. In section 6.1 we mentioned that the exam type might have an influence on the CEFR levels assigned by the human raters. Raters might form expectations about the essays at certain exam types, that might influence the ratings. All other considerations set aside, it is common that most people that take a test for a certain level, also achieved this level - more people pass an exam than fail it.

The experiment was conducted with 10 fold cross validation on the whole data set and with all features. The resulting classifier predicted the exam type with an accuracy of 88.9 % and an F-measure of 88 %. This indicates that our feature set can more accurately predict the exam type than the essay rating level. It cannot be concluded from this experiment however, as the conditions are not equal. For instance, the class distribution is different. While the exam type classifier has an equal amount of data for each class, the essay rating level classifier was trained on a skewed data set, which could be one reason that the essay rating level classifier performed worse. We leave the investigation of the exact reasons for future work.

7 Conclusion

During the initial phase of this work, we explored the *MERLIN* data set and identified some interesting characteristics and challenges, in particular the high amount of spelling errors and frequently spurious punctuation. We addressed the spelling errors by integrating *Google Spell Check* into our preprocessing pipeline. We surveyed methods for sentence boundary detection that, unlike conventional tools, do not mainly rely on punctuation. Finally we implemented a tool for automatically identifying missing sentence boundaries. Although the results seemed promising, the tool was not reliable enough to use it in this work.

We learned that the texts in the *MERLIN* corpus vary immensely in length between the CEFR levels (exam types as well as essay rating levels). There are roughly the same number of texts for each of the CEFR exam types

A1-C1, but not for the actual CEFR levels assigned by the human raters. As the ratings represent the actual proficiency, the data set for our proficiency classifier did not have an even class distribution.

We implemented a wide range of theory driven lexical, syntactic and morphological features as well as data driven tense and parse rule features. They were mainly inspired by previous research on proficiency assessment, essay grading, readability classification and SLA studies. This resulted in a feature set of more than 3000 features.

We addressed proficiency assessment as a classification problem, using each of the five essay rating levels as one class. SMO was chosen for building our classifiers. We trained one classifier for each feature group to find the most predictive group. The lexical classifier performed best (holdout 60.5 %, CV 68.7 % accuracy), followed by the morphological and syntactic classifiers. The language model, tense and parse rule features performed rather poorly. However, the tense and parse rule features achieved much better results when represented as binary instead of frequency weighted vectors (+18 % and +7 % accuracy respectively).

When combining several feature groups we found out that combinations of two or three groups mostly performed better than one group alone. Particularly, the lexical and morphological group (holdout 61.1 %, CV 69.4 % accuracy) seemed to complement to each other. The best three class combination consisted of the lexical, morphological and language model groups (holdout 61.1 %, CV 70.1 % accuracy). When combining more than three groups, the classifiers' performances dropped. We attribute this to the accumulation of too many redundant features. A supplementary statistical analysis showed that particularly in the data driven classes there are many features with a correlation below 0.1. On the other hand there were hardly any high ($r > 0.7$) intercorrelations between features belonging to different groups.

We applied correlation based feature selection on the best combinations of three and four feature groups and on the whole feature set. The best model (holdout 62.7 %, CV on training set 72.5 % accuracy) contained 34 features from the syntactic, lexical, language model and morphological groups. Interpreting the model showed that it contained most of the linguistic aspects that are encoded in our entire feature set.

We experimented with a two class "pass or fail" scenario for each of the exam types. For most exam types there were significantly more "pass" than "fail" samples. We showed that cost sensitive learning can be used to simulate a more equal class distribution or to give more priority to detecting the "fail" class.

Additional experiments showed that our classifier performed better when

using RASCH corrected scores for training and testing (67.9 % compared to 64.5 % accuracy). This implies that some of the classifier errors might be due to inconsistent human ratings in the original data set. Finally we used our feature set to train a classifier for predicting the exam type. Strikingly, it performed better than our proficiency classifier (88.9 % compared to 64.5 % accuracy). The implications of this could not be examined further in the course of this work, but will be an interesting topic for further research.

In summary it can be said that we gained interesting insights not only into the performance of particular features, but also into the issues associated with the task and the data set. As this was a first attempt at language proficiency classification for German there is no previous work that we could directly compare our work to. However, the performance of our classifiers seemed promising.

8 Perspectives for Future Work

We conducted a wide range of experiments and many of our results led to new questions. With respect to machine learning it would be interesting to investigate, why different combinations of feature groups performed best in the holdout and in the cross validation scenario. It was mentioned in section 6.4 that some features might encode text length rather than language proficiency. For clarification, the experiments in this work should be repeated with samples that are controlled for text length. It is also worth investigating why our feature set seemed to be more useful for detecting the exam type than the essay rating level.

If the system was going to be used in a real life scenario, it would be necessary to test its robustness against score manipulation by students who figured out the criteria on which the essay ratings are assigned. However, since the number of our criteria is rather large, we think that tricking the system would be hard.

Furthermore, it would be desirable to experiment with different ways of handling the uneven class distribution in the multiclass scenario. Also, more sophisticated machine learning techniques such as ensemble classifier should be employed. For example, if one classifier was built for each group of features and then combined into an ensemble, the tense and parse rule groups could be used as binary features.

In addition to more machine learning experiments a thorough statistical analysis would help to arrive at a better understanding of the data set and might reveal the influence of exam type, L1 background or effect of different tasks on the essay rating level.

Finally the tool for missing sentence boundary detection could be further developed.

References

Naoki Abe, Bianca Zadrozny, and John Langford. An iterative method for multi-class cost-sensitive learning. In *In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 3–11, Florida, USA, 2004. AAAI Press.

Yigal Attali and Jill Burstein. Automated essay scoring with e-rater v.2. *Journal of Technology, Learning, and Assessment*, 4(3), February 2006. URL `http://escholarship.bc.edu/cgi/viewcontent.cgi?article=1049&context=jtla`.

R. H. Baayen, R. Piepenbrock, and L. Gulikers. The CELEX lexical database (cd-rom). CDROM, `http://www.ldc.upenn.edu/Catalog/readme_files/celex.readme.html`, 1995.

R. Harald Baayen. *Analyzing Linguistic Data. A Practical Introduction to Statistics using R*. Cambridge University Press, 2008.

Olaf Bärenfänger. Assessing the reliability and scale functionality of the merlin written speech sample ratings. Technical report, European Academy, Bolzano, Italy, 2012.

Gunnar Bech. *Studien über das deutsche verbum infinitum*. Historisk-filologiske Meddelelser udgivet af Det Kongelige Danske Videnskabernes Selskab. Bind 35, no. 2, 1955; Bind 36, no. 6, 1957; Kopenhagen, 1955. Reprinted 1983, Tübingen: Max Niemeyer Verlag.

Jared Bernstein, John De Jong, David Pisoni, and Brent Townshend. Two experiments on automatic scoring of spoken language proficiency. In *Proceedings of InSTIL2000 (Integrating Speech Tech. in Learning)*, pages 57–61, Dundee, Scotland, 2000.

D. Biber, B. Gray, and K. Poonpon. Should we use characteristics of conversation to measure grammatical complexity in l2 writing development? *TESOL QUARTERLY*, 54:5–35, 2011.

Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 89–97, Beijing, China, 2010.

E. Briscoe, J. Carroll, and R. Watson. The second release of the rasp system. In *In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia, 2006.

Ted Briscoe, B. Medlock, and O. Andersen. Automated assessment of esol free text examinations. Technical report, University of Cambridge Computer Laboratory, 2010.

Stanley F. Chen and Joshua T. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13: 359–394, October 1999.

Kevyn Collins-Thompson and Jamie Callan. A language modeling approach to predicting reading difficulty. In *Proceedings of HLT/NAACL 2004*, Boston, USA, 2004. URL `http://www.cs.cmu.edu/~callan/Papers/hlt04-kct.pdf`.

Scott Crossley, Tom Salsbury, and Danielle McNamara. Measuring l2 lexical growth using hyperniymic relationships. *Language Learning*, 59:307–334, 2009.

Scott Crossley, Tom Salsbury, and Danielle McNamara. The development of polysemy and frequency use in english second language speakers. *Language Learning*, 60:573–605, 2010.

Scott A. Crossley, Tom Salsbury, and Danielle S. McNamara. Predicting the proficiency level of language learners using lexical indices. In *Language Testing*, 2011a.

Scott A. Crossley, Tom Salsbury, Danielle S. McNamara, and Scott Jarvis. Predicting lexical proficiency in language learners using computational indices. *Language Testing*, 28:561–580, 2011b.

Chris Culy, Corina Dima, and Emanuel Dima. Through the looking glass: Two approaches to visualizing linguistic syntax trees. presented at IV2012, July 2012.

Febe de Wet, Christa van der Walt, and Thomas Niesler. Automatic large-scale oral language proficiency assessment. In *INTERSPEECH 2007, 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium*, 2007.

Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. Read-it: Assessing readability of italian texts with a view to text simplification. In

Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies, pages 73–83, 2011.

Semire Dikli. An overview of automated scoring of essays. *The Journal of Technology, Learning, and Assessment (JTLA)*, 5(1):4–35, August 2006. URL `http://escholarship.bc.edu/cgi/viewcontent.cgi?article=1044&context=jtla`.

Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pages 155–164, San Diego, CA, USA, 1999.

B. Favre, D. Hakkani-Tr, S. Petrov, and D. Klein. Efficient sentence segmentation using syntactic features. In *Spoken Language Technology Workshop (SLT)*, 2008.

Lijun Feng. *Automatic Readability Assessment*. PhD thesis, City University of New York (CUNY), 2010. URL `http://lijun.symptotic.com/files/thesis.pdf?attredirects=0`.

Thomas Francois and Cedrick Fairon. An ai readability formula for french as a foreign language. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.

Peter Gallmann and Horst Sitta. *Deutsche Grammatik*. Interkantonale Lehrmittelzentrale. Lehrmittelverlag des Kantons Zrich., 2004.

Y. Gotoh and S. Renals. Sentence boundary detection in broadcast speech transcripts. In *Proceedings of the International Speech Communication Association (ISCA) Workshop: Automatic Speech Recognition: Challenges for the New Millennium (ASR-2000)*, pages 228–235, Paris, 2000.

Arthur C. Graesser, Danielle S. McNamara, Max M. Louweerse, and Zhiqiang Cai. Coh-metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments and Computers*, 36:193–202, 2004. URL `http://home.autotutor.org/graesser/publications/bsc505.pdf`.

Günther Grewendorf, Fritz Hamm, and Wolfgang Sternefeld. *Sprachliches Wissen. Eine Einführung in moderne Theorien der grammatischen Beschreibung*. Surhkamp, 1989.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. In *The SIGKDD Explorations*, volume 11, pages 10–18, 2009.

Julia Hancke, Detmar Meurers, and Sowmya Vajjala. Readability classification for german using lexical, syntactic, and morphological features. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 1063–1080, Mumbay, India, 2012.

Peter Harrington. *Machine Learning in Action*. Manning Publications Co., 2012.

John A. Hawkins and Paula Buttery. Using learner language from corpora to profile levels of proficiency – Insights from the English Profile Programme. In *Studies in Language Testing: The Social and Educational Impact of Language Assessment*. Cambridge University Press, Cambridge, 2009.

John A. Hawkins and Paula Buttery. Criterial features in learner corpora: Theory and illustrations. *English Profile Journal*, 2010.

Michael Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-07)*, pages 460–467, Rochester, New York, 2007.

Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications at ACL-08*, Columbus, Ohio, 2008.

J. Heister, K.-M. Würzner, J. Bubenzer, E. Pohl, T. Hanneforth, A. Geyken, and R. Kliegl. dlexdb - eine lexikalische datenbank fr die psychologische und linguistische forschung. *Psychologische Rundschau*, 62:10–20, 2011.

Verena Henrich and Erhard Hinrichs. Gernedit - the germanet editing tool. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

Kellogg W. Hunt. Grammatical structures written at three grade levels. NCTE Research Report No. 3, 1965.

Kellogg W. Hunt. Do sentences in the second language grow like those in the first? *TESOL Quarterly*, 4(3):195–202, 1970.

J. P. Kincaid, R. P. Jr. Fishburne, R. L. Rogers, and B. S Chissom. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease formula) for Navy enlisted personnel. Research Branch Report 8-75, Naval Technical Training Command, Millington, TN, 1975.

T. Landauer, D. Laham, and P. Foltz. Automated scoring and annotation of essays with the intelligent essay assessor. In M.D. Shermis and J. Burstein, editors, *Automated Essay Scoring: A cross-disciplinary perspective*, pages 87–112. Law, 2003.

Roger Levy and Galen Andrew. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *5th International Conference on Language Resources and Evaluation*, Genoa, Italy, 2006.

Charles X. Ling and Victor S. Sheng. Class imbalance problem. In *Encyclopedia of Machine Learning*, page 171. 2010.

Y. Liu, A. Stolcke, E. Shriberg, and M. Harper. Using conditional random fields for sentence boundary detection in speech. In *In Proceedings of ACL*, pages 451–458.

Y. Liu, A. Stolcke, Shriberg E., and Harper M. Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speec. In *In Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing*, pages 64–71, 2004.

Xiaofei Lu. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474–496, 2010.

Xiaofei Lu. The relationship of lexical richness to the quality of ESL learners' oral narratives. *The Modern Languages Journal*, 2012.

Christopher Manning and Hinrich Schütze. *Foundations of statistical Natural Language Processing*. MIT, 1999.

P. M. McCarthy and S. Jarvis. A theoretical and empirical evaluation of vocd. *Language Testing*, 24:459–488, 2007. URL `https://umdrive.memphis.edu/pmmccrth/public/Papers/080767_LTJ_459-488.pdf?uniq=czcyb`.

Philip McCarthy and Scott Jarvis. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42(2):381–392, 2010. ISSN 1554-351X. doi: 10.3758/BRM.42.2.381. URL `https://serifos.sfs.uni-tuebingen.de/svn/resources/trunk/papers/McCarthy.Jarvis-10.pdf`.

Walt Detmar Meurers. *Lexical Generalizations in the Syntax of German Non-Finite Constructions*. Phil. dissertation, Eberhard-Karls-Universität Tübingen, 2000. URL `http://www.sfs.uni-tuebingen.de/~dm/papers/partI-2up.pdf`. Published as: Arbeitspapiere des SFB 340, Nr. 145.

George Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, November 1995. URL `http://aclweb.org/anthology/H94-1111`.

Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

R. Nagata, J. Kakegawa, and T. Kutsuwa. Detecting missing sentence boundaries in learner english. In S. Bolasco, I. Chiari, and L. Giuliano, editors, *Statistical Analysis of Textual Data. Proceedings of the 10th International Conference (JADT 2010).*, volume Volume 2, 2010.

S. Palmer and M. Hearst. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23:241–267, 1997.

Sarah E. Petersen and Mari Ostendorf. A machine learning approach to reading level assessment. *Computer Speech and Language*, 23:86–106, 2009.

Anna N. Rafferty and Christopher D. Manning. Parsing three German treebanks: lexicalized and unlexicalized baselines. In *Proceedings of the Workshop on Parsing German*, PaGe '08, pages 40–46, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1621401.1621407`.

Marga Reis. Bilden modalverben im deutschen eine syntaktische klasse? *Modalitt und Modalverben im Deutschen / hrsg. von Reimar Müller*, 2001.

Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, pages 16–19, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics. doi: 10.3115/974557.974561. URL `http://dx.doi.org/10.3115/974557.974561`.

Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, Dublin, Ireland, 1995. URL `http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger2.pdf`.

Helmut Schmid and Florian Laws. Estimation of conditional probabilities with decision trees and an application to fine-grained pos tagging. In *COLING '08 Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1, pages 777–784, Stroudsburg, PA, 2008. Association for Computational Linguistics. URL `http://www.ims.uni-stuttgart.de/projekte/gramotron/PAPERS/COLING08/Schmid-Laws.pdf`.

Bernhard Schölkopf and Alexander J. Smola. A short introduction to learning with kernels. In *Advanced Lectures on Machine Learning*. Springer-Verlag, 2003.

Sarah Schwarm and Mari Ostendorf. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 523–530, Ann Arbor, Michigan, 2005. doi: http://dx.doi.org/10.3115/1219840.1219905.

Wolfgang Seeker and Jonas Kuhn. Making ellipses explicit in dependency conversion for a german treebank. In *In Proceedings of the 8th International Conference on Language Resources and Evaluation, 31323139. Istanbul, Turkey: European Language Resources Association (ELRA)*, 2012.

Luo Si and Jamie Callan. A statistical model for scientific readability. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM)*, pages 574–576. ACM, 2001.

Advaith Siddharthan. An architecture for a text simplification system. In *In Proceedings of the Language Engineering Conference 2002 (LEC 2002)*, 2002.

Wojciech Skut, Brigitte Kreen, Torsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of the Fith Conference on Applied Natural Language*, Washington, D.C., 1997. URL `http://www.coli.uni-sb.de/publikationen/softcopies/Skut:1997:ASF.pdf`.

A. Stolcke and E. Shriberg. Automatic linguistic segmentation of conversational speech. In *Proceedings of ICSLP*, 1996.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904, Denver, USA, 2002. URL `http://www.speech.sri.com/cgi-bin/run-distill?papers/icslp2002-srilm.ps.gz`.

Sowmya Vajjala and Kaidi Loo. Role of morpho-syntactic features in estonian proficiency classification. In *Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications (BEA8), Association for Computational Linguistics*, 2013.

Sowmya Vajjala and Detmar Meurers. On improving the accuracy of readability classification using insights from second language acquisition. In Joel Tetreault, Jill Burstein, and Claudial Leacock, editors, *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA7) at NAACL-HLT*, pages 163–173, Montral, Canada, June 2012. Association for Computational Linguistics. URL `http://aclweb.org/anthology/W12-2019.pdf`.

Tim Vor der Brück and Sven Hartrumpf. A semantically oriented readability checker for german. In *Proceedings of the 3rd Language & Technology Conference*, 2007.

Tim Vor der Brück, Sven Hartrumpf, and Hermann Helbig. A readability checker with supervised learning using deep syntactic and semantic indicators. *Informatica*, 32(4):429–435, 2008.

Katrin Wisniewski, Andrea Abel, and Detmar Meurers. Merlin. multilingual platform for the european reference levels: Interlanguage exploration in context. eu lifelong learning project 518989-llp-1-2011-1-de-ka2-ka2mp. eu llp proposal. Proposal, 2011.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Amsterdam; Boston, MA, 2nd edition, 2005.

Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=645526.657137`.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th*

Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 180–189, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL http://dl.acm.org/citation.cfm?id=2002472.2002496. Corpus available from http://ilexir.co.uk/applications/clc-fce-dataset.

Heike Zinsmeister, Marc Reznicek, Julia Ricart Brede, Christina Rosn, and Dirk Skiba. Wissenschaftliches netzwerk: Annotation und analyse argumentativer lernertexte. konvergierende zugange zu einem schriftlichen korpus des deutschen als fremdsprache. wissenschaftliches netzwerk kobalt-daf, dfg antrag. Proposal, 2011.