Get Semantic With Me! The Usefulness of Different Feature Types for Short-Answer Grading

Ulrike Padó
Hochschule für Technik Stuttgart
Schellingstr. 24
70174 Stuttgart
ulrike.pado@hft-stuttgart.de

Abstract

Automated short-answer grading is key to help close the automation loop for large-scale, computerised testing in education. A wide range of features on different levels of linguistic processing has been proposed so far. We investigate the relative importance of the different types of features across a range of standard corpora (both from a language skill and content assessment context, in English and in German). We find that features on the lexical, text similarity and dependency level often suffice to approximate full-model performance. Features derived from semantic processing particularly benefit the linguistically more varied answers in content assessment corpora.

1 Introduction

Computerised testing is becoming ubiquitous in the educational domain, and automated and semi-automated grading of tests is in high demand to relieve the workload of teachers (especially in the context of Massive Open On-line Courses or repeated testing for continuous feedback during the academic year). NLP is a key technology to close or at last narrow the automation loop for grading of free-text answers and essays. We focus on the automated grading of *short-answer questions* (i.e., assessment questions that require a free-text answer up to two or three sentences in length). For this task, the training data consists of a question, at least one reference answer and several student answers. Systems then predict answer accuracy as a binary *correct-incorrect* decision or as a more fine-grained multi-class problem (or even a regression task predicting points). In contrast to the related essay grading task, correct student answers stay closer to the reference answer than good essays might to an example essay.

As is often the case in young research areas, an important contribution was made by the Semeval-2013 shared task (Dzikovska et al., 2013), which introduced standard evaluation benchmarks. Two large data sets are now available with performance standards for system comparison.

On the shared task data, researchers have experimented with various features based on linguistic processing, from syntactic information (used by a majority of entries in SemEval-2013, Dzikovska et al. (2013)) to deep semantic representations (Ott et al., 2013) or Textual Entailment (TE) systems (Zesch et al., 2013). Others, staying closer to the surface level, have recently experimented with sophisticated measures of textual similarity (Jimenez et al., 2013; Sultan et al., 2016) or inferring informative answer patterns (Ramachandran et al., 2015). However, similar to other NLP tasks (like, for example, TE), one of the biggest challenges remains beating the lexical baseline: At SemEval-2013, the baseline consisting of textual similarity measures comparing reference and student answer frequently was not outperformed.

Given the large feature space proposed so far and the lack of consensus about where to find the most useful features, we ask whether there are any regularities in the predictiveness of features across corpora. Is there a hierarchy of features that are always, sometimes or never useful across different corpora and languages? Are features from deep linguistic processing informative over and above the lexical baseline, or are they subsumed by the more shallow features? And, finally, do the optimal feature sets differ with corpus characteristics like language or elicitation task?

We will investigate these questions as follows: Section 2.1 defines our task and hypotheses. Section 2.2 introduces the corpora and 3 describes the features. Section 4 specifies our implementation of the experiments. We first validate our feature set against literature results in Section 5, then look at feature predictiveness individually in Section 6 and in combination in Section 7. Section 8 concludes.

2 Background: Task and Data

We look for highly predictive features for the short-answer grading task that generalise over different corpora. We also ask how much information can be drawn from more abstract features from deeper levels of linguistic processing that is not covered by the strong NGram and text similarity baselines.

2.1 Task and Hypotheses

We investigate these questions by looking at unseen-question 2-way classification of short answers. In this task, the test data contains only questions (and the corresponding answers) not seen during training, but from a similar domain as the training data. We choose this task because it makes no assumptions about pre-existing student answers for each question, which maps well to small-scale testing in many educational settings. The task is binary classification of answers as *correct* or *incorrect*.

We select data from the spectrum of available short-answer corpora (see, e.g., the excellent overview article by Burrows et al. (2015)) according to two criteria: Language and elicitation task. There are two predominant tasks: On the one hand, there are corpora that assess content mastery in specific knowledge domains. These corpora contain answers by mostly highly proficient speakers. On the other hand, there are learner corpora assessing language students' reading comprehension by asking questions about the content of a text. Answers to these questions are characterised by learner mistakes and the heavy influence of *lifting* answers from the reading text, with the result of overall less variation within answers. In order to vary language, we use one German and one English corpus for each mode (see 2.2 below).

We hypothesise that the higher levels of answer variation in content-assessment corpora as opposed to the language-skill corpora will necessitate features from deeper processing levels to uncover parallels between student and reference answer. We do not expect corpus language to have a big effect, except perhaps in the usefulness of pre-processing like lemmatisation for inflection-rich German.

2.2 Data

We use the corpora listed in Table 1. The SciEntsBank (SEB) and Beetle corpora are the SemEval-2013 corpora (Dzikovska et al., 2013), which we consider as one data set. Both contain content assessment questions from science instruction, in English. CSSAG is a set of German content assessment questions about programming in Java; the data set used here is an extension of the corpus described in Padó and Kiefer (2015), following the same design principles. CREG (Meurers et al., 2011b) and CREE (Meurers et al., 2011a) are language-skill corpora. Learners of German and English, respectively, read texts and answered questions about their content.

	#Questions/ #Answers	#Q/#A (Test Set)	Task	Language
SEB (Dzikovska et al., 2013)	135/4969	16/733	Content	English
Beetle (Dzikovska et al., 2013)	47/3941	9/819		
CSSAG (Padó and Kiefer, 2015)	31/1926	NA		German
CREG (Meurers et al., 2011b)	85/543	NA	Language	
CREE (Meurers et al., 2011a)	61/566	NA		English

Table 1: Corpus sizes and characteristics

For Beetle and SEB, ample test sets exist which we use in Section 5 to validate our full models before delving into feature analysis. For the smaller corpora, there are no separate test sets[1] and the data sets

[1] There is a designated test set for CREE, but it repeats some questions from the training set, so it is not appropriate for the unseen question task. We therefore combined development and test data for CREE.

were considered too small to create them. Following Hahn and Meurers (2012), we present results for leave-one-question-out cross-validation, where we hold out each question in turn and train models on the remaining data.

All corpora contain the question texts, at least one reference answer per question and the student answers with a human-assigned correctness judgment. All corpora have explicit *correct/incorrect* annotation, except for CSSAG. CSSAG answers are scored in half-point steps up to a maximum number of points per question (usually 1 or 2). We convert CSSAG scores into binary labels by mapping all answers with more than 50% of points to *correct* and all other answers to *incorrect*.

3 Features

We compute established literature features on five different levels of linguistic processing. In the order of processing complexity, these are NGram features, text similarity features, dependency features, abstract semantic representations in the LRS formalism (Richter and Sailer, 2004) and entailment votes from a TE system, which we treat as a black box.

In the unknown question setting, all features are computed in relation to the reference answer given for each question (the question is also considered for some features, see below). Features usually code the overlap between units (NGrams, dependency relations, etc.) in the reference and student answer. We use the reference answer as the basis, so the features express the percentage of reference answer units shared between student and reference answer. The higher the percentage, the more completely does the student answer cover the reference. If the percentage is lower, the student answer is probably incomplete. The inverse percentage can of course also be computed; where the corresponding features performed well, we include them also. Wherever there is more than one reference answer, we use the maximum overlap of all the answer options, assuming that graders will evaluate student answers according to the most similar reference answer.

Table 2 gives an overview over the feature set. In more detail, the features are:

NGram features measure the overlap in uni-, bi- and trigrams between reference and student answer. NGrams are computed on both tokens and lemmas (to raise coverage).

Similarity measures compare reference and student answer on the text level. We use Greedy String Tiling (GST, Wise (1996)), a string-based algorithm popular in plagiarism detection that deals well with insertions, deletions and re-arrangement of the text.[2] We also use the classical Cosine measure as a vector-based approach and compute the Levenshtein edit distance between the texts. The measures are run on lemmatised text before (with stop words, WSW) and after stop word filtering (SWF). Stop word filtering includes removal of words in the question (*question word demotion*, Mohler et al. (2011)). The rationale is that students should be graded on the new information they provide over and above the concepts mentioned in the question. We chose not to use similarity measures that need external resources (such as WordNet or large corpora), since they may not be equally appropriate for the different corpus domains and show inconsistent performance.

Dependency features code the overlap between the student and reference answer dependency relations in terms of lemmatised triples of governor, dependency type and dependent.

Semantics features are derived by the parsing and alignment component in CoSeC (Hahn and Meurers, 2012). It constructs LRS (Lexical Resource Semantics, Richter and Sailer (2004)) analyses of the texts and attempts to align the components. We then compute the overlap in aligned components between reference and student answers as well as the question and student answer. The motivation for the latter measure is similar to question-word demotion in that high overlap between question and answer may point to question copying with little additional content.

TE decisions are computed using the Excitement Open Platform[3] (EOP, Magnini et al. (2014)). Dzikovska et al. (2013) propose constructing the Text from question and student answer and using

[2] Minimum string length is four characters.

[3] `http://hltfbk.github.io/Excitement-Open-Platform/`

the reference answer as the Hypothesis that may or may not be entailed by the Text. For us, this led to many false-positive entailment judgments by the TE system, most likely because the longer the Text, the easier it becomes to construct relations between Text and Hypothesis. We therefore use only the student answer as the Text. Our features are the entailment decision itself and the confidence score returned by the system. If there are multiple reference answers, the student answer may well entail one, but not the others. Therefore, we record any Entailment decision and its confidence score over any Non-Entailment decision, and in the case of only Non-Entailment decisions, record the lowest confidence score to capture the judgment closest to Entailment. This means that a high score size correlates with a positive decision and a low score with a negative decision.[4]

Feature Group	Feature Names
NGram	Unigram(Token,Lemma), Bigram(Token,Lemma), Trigram(Token,Lemma)
Similarity	GST(WSW,SWF), Cosine(WSW,SWF), Levenshtein(WSW,SWF)
Dependency	SRDependency, RSDependency
Semantics	LRS-QS, LRS-RS, LRS-SR
TE	TEDecision, TEConfidence

Table 2: Overview of the feature set

4 Method

We pre-processed the corpora with the DKPro pipeline (Eckart de Castilho and Gurevych, 2014), using the OpenNLP segmenter[5], the TreeTagger for POS tags and lemmas (Schmid, 1995) and the MaltParser (Nivre, 2003) for dependency parses. All tools (including the LRS parser and the EOP TE system) are used as-is without additional evaluation and tuning on our data.

Since our goal is to gain insight into the contribution of the different feature groups, we consider only one learning algorithm and do not investigate ensemble learning (although this is a common and promising approach in the literature (Dzikovska et al., 2013)). For our small data sets, overfitting is a concern. We therefore use decision trees, namely the J48 implementation in the Weka machine learning toolkit (Hall et al., 2009), which addresses overfitting by a pruning step built into the algorithm.

We report unweighted average F1 scores for comparability with Dzikovska et al. (2013), and, for the full models, accuracy for comparison to Hahn and Meurers (2012) and Meurers et al. (2011a). Tests for significance of differences between results are carried out by stratified shuffling (Yeh, 2000). The independent observations needed for this approach are the sets of answers belonging to one question.

5 Full Models and Literature Benchmarks

As the first step, we compare the performance of the decision tree algorithm and the whole feature set to the literature benchmarks for the data sets. We show that the model and features we chose achieve realistic performance to ensure that our analyses below are meaningful.

In addition to the benchmarks, we report the frequency baseline (always assign the more frequent class) and a lexical baseline (a decision tree trained with just the UnigramToken feature)[6].

For the binary grading task, the human upper bound for accuracy (measured as agreement between the raters) is in the high eighties. For the CREE and CREG corpora, grader agreement is reported as 88% (Bailey and Meurers, 2008) and 87% (Ott et al., 2012), respectively.

Table 3 lists the unweighted average F1 scores and accuracies for the different data sets. The SEB and Beetle figures are for the held-out test sets; for the other data sets, we report leave-one-question-out cross-validation results. All models outperform the frequency baseline.

[4] We use the MaxEntClassification algorithm with settings Base+WN+TP+TPPos+TS for English and settings Base+GNPos+DBPos+TP+TPPos+TS for German.

[5] `https://opennlp.apache.org/`

[6] Note that this baseline differs from the lexical baseline used in the SemEval-2013 evaluation, where a combination of similarity measures was used. The SemEval-2013 lexical baseline is the same for SEB and F=78.8 for Beetle.

	SEB		Beetle		CSSAG		CREG		CREE	
	F	Acc	F	Acc	F	Acc	F	Acc	F	Acc
Frequency Bsl	37.1*	59.0	36.7*	58.0*	38.4*	62.4*	37.4*	52.7*	44.4*	59.5*
UnigramToken Bsl	61.8*	64.0	69.8	72.9	67.4	72.0	78.6	78.6	66.5	81.3
Full model	52.8	61.5	68.1	70.8	66.6	69.3	81.5	82	70.2	80.4
Literature	62.9	–	66.6	–	–	–	–	86.3	–	88.4

Table 3: Performance of the full feature set in comparison to baselines and literature results. * indicates a significant difference between baseline and full model.

Four of the five models do not significantly differ from the UnigramToken baseline, although two numerically outperform it. This is a familiar picture from the SemEval-2013 competition and underscores the difficulty of the task.

A notable anomaly is the SEB model, which significantly underperforms on F-score and numerically underperforms on accuracy against both baselines. The leave-one-question-out cross-validation result for this model on the training set is comparable to the Beetle test set result at an F-score of 66.0 and accuracy of 67.7. We hypothesise that the training and test set for the SEB data differ substantially.

The SEB model performance of course also does not reach the literature result (although the cross-validation result of F=66.6 is comparable), while the Beetle model even numerically outperforms the literature benchmark (we compare to the median participant performance at SemEval-2013, Dzikovska et al. (2013)). For CSSAG, no prior literature results exist. The CREG result is roughly similar to the best model to date reported in Hahn and Meurers (2012). The literature result for CREE (Meurers et al., 2011a) is not completely comparable, as it was computed on the held-out test set that does not satisfy the unseen question task. Therefore, it is not surprising that our model does somewhat worse on a purely unseen question evaluation.

Overall, with the exception of the SEB model, we have been able to verify that out feature set and learner are able to approximate state-of-the-art results. We still include the SEB data set in our analyses below since the leave-one-question-out cross-validation result is much more consistent with the other models and we hypothesise a mismatch of test and training data.

6 Performance of Individual Features

For our analysis of feature impact, we first look at the performance of each feature individually. We train a decision tree with just that feature and report unweighted average F1 scores. We present only features that outperform the frequency baseline by at least 10 points F-score. The cells in Table 4 show the difference in F-score between the single-feature and full-model performance. Features that perform numerically close to the full model (within 15 percent of the F-score) are bold-faced.

We first discuss the table from the point of view of the different feature groups. As expected, the **NGram features** are strong and approximate full-model performance consistently. The higher-order NGrams drop off against the Unigrams since they are sparser. The lemmatised NGram features were introduced to potentially overcome this problem, but they consistently do less well than the token-level features. Analysis shows that lemmatisation yields higher overlap percentages between reference and student answer, but this figure now correlates less with answer accuracy. Apparently, there are important differences between reference and student answer on the token level that are lost through lemmatisation.

The **similarity measures** are also strong across the board. Among the measures we tested, Greedy String Tiling is the best predictor of response accuracy. Further, our results support the suggestion by Okoye et al. (2013) that stop words should not be removed, but we can qualify this recommendation: For measures like Greedy String Tiling and Levenshtein that explicitly operate on word sequences, stop word removal hurts performance. For the Cosine measure, on the other hand, removal is generally beneficial because it removes spurious overlap. Levenshtein edit distance is the least predictive of the similarity measures. This fits well with the analysis in Heilmann and Madnani (2013), who also see uneven performance of the model containing their edit-distance feature.

Feature Group	Feature	SEB	Beetle	CSSAG	CREG	CREE
NGrams	UnigramToken	**-4.3**	**-1.8**	**0.8**	**-2.9**	**-3.7**
	BigramToken	**-8.2**	**-1.4**	**-5.6**	**0**	–
	TrigramToken	-16.5	**-4.8**	-20.7	**-8.7**	–
	UniLemma	-21.4	-12.6	**-3.6**	-12.3	**-9.7**
	BiLemma	-12.9	**-6.1**	–	**-10.2**	-14.4
	TriLemma	–	-15.1	–	-20.7	–
Similarity	GST-WSF	-17.1	**-9**	**1.1**	**-10.7**	**-6.2**
	CosineWSF	-12.5	**-9.9**	–	**-11.5**	–
	LevenshteinWSF	–	-16.2	**0.6**	-29.8	–
	GST-WSW	**-7.7**	**-8**	**-1.4**	**-11.8**	**–2.2**
	CosineWSW	-15.1	**-9.1**	–	-22.2	–
	LevenshteinWSW	–	-20.8	**1**	-22.8	–
Dependency	RSDependency	**-7.5**	**-7**	**-9**	**-10.5**	–
	SRDependency	-13.5	-13.1	**-4.9**	–	–
Semantics	LRS-QS	–	**-9.3**	-16.6	–	–
	LRS-RS	-13.6	**-1.6**	**-3.4**	-24.9	–
	LRS-SR	-16.2	-13.7	–	-23.8	–
TE	TEDecision	–	–	-19.1	-34.3	-15.6
	TEConfidence	-12.9	-13.3	**-4.6**	**-10.7**	**2.2**
	Full model	66.0	72.6	66.6	81.5	70.2

Table 4: Performance of individual features across all data sets ($F_{feature} - F_{full\ model}$ for all $F_{feature}$ at least 10 points F-score above the Frequency baseline).

The **dependency** features are also informative for all corpora (except for CREE). Here and in the semantic features, we again find that the RS normalisation works better than the SR normalisation, that is, specifying how much of the reference answer is covered by the student answer predicts overall accuracy better than looking at how much of the student answer is present in the reference answer. This is because the latter direction does not accurately model incomplete student answers.

The **semantic representations** are highly predictive, but only for Beetle and CSSAG, although they are within 20% F-score for SEB. The overlap between question and student answer (QS) is probably informative for Beetle because of questions that ask about specific components in an electric circuit which have to be mentioned in a correct answer. This observation calls into question the usefulness of question word demotion in all situations. Specifically in Beetle, question word demotion can be counterproductive. Take, for example, the question *Why do you think those terminals and the negative battery terminal are in the same state?* with one reference answer *Terminals 1, 2 and 3 are connected to the negative battery terminal*, and its demoted version *1, 2 3 connected to*, which matches correct answers as well as incorrect answers which speak about a connection to the positive battery terminal.

The **TE** confidence feature works well for CSSAG and outperforms the full model for CREE by two points F-score. Performance for SEB and Beetle is within 20% F-score of the full model performance. Recall that the construction of the confidence value guarantees a correlation of high confidence with an Entailment decision and lower confidence with a Non-Entailment decision, so the feature carries most of the information of the nominal TEDecision feature in addition to the graded confidence.

In sum, all features are useful, but not in all cases. As expected, there are workhorse features like NGrams and similarity that strongly predict response accuracy across the board. We find that this general applicability extends to dependency features, as well. The more abstract semantic and TE features are useful in specific cases.

To further analyse these performance patterns, we now turn to an analysis from the point of view of the corpus types. We hypothesised in Section 2.1 that there would be little influence of language, but a noticeable difference between the corpora collected with different elicitation tasks.

First of all, there is indeed no discernible difference between the German and English corpora. Even NGram lemmatisation, while helpful for CSSAG and CREG, also approximates the full models for Beetle and CREE quite well and the best token-level NGram features always outperform the lemmatised features.

We do however see a clear difference between the content assessment and language-skill assessment corpora. The language-skill corpora (CREG and CREE) profit comparatively little from the deeper processing of the dependency and LRS features; NGram and text similarity features are however very strong predictors of response accuracy. This can be explained by the typical answers in these corpora: Since students' language proficiency is limited, they often lift all or part of their answers directly from the reading. The target answers are adapted from the same texts and therefore lexically similar. Lexical and string overlap are therefore sufficient to distinguish between a correct and an incorrect answer. Then why are the TE features so strikingly successful for these corpora? This can be explained by the fact that, for CREG, and especially for CREE, the reference answers are slightly re-formulated from the reading texts by the instructor, a highly proficient speaker of the target language (frequent changes include tense and contractions, but also paraphrasing, often with POS changes for the words involved). The generalisation strategies in the TE system help find the underlying semantic similarity that is obscured on the token level.

For the content assessment corpora, in addition to the NGram and similarity features, features from deeper levels of processing are always useful. (The TE confidence is within 20% F-score for SEB and Beetle, as are the "missing" semantic and dependency features.) The deeper processing levels apparently help uncover paraphrasing by (mostly) language-proficient test-takers.

From the analysis of individual feature performance, we thus find that the NGram and similarity surface features, as well as the dependency features, are predictive for every corpus, but the features from deeper processing are useful especially for the content assessment corpora.

7 Feature Groups and Combined Performance

The discussion in Section 6 showed a clear performance pattern for the different feature groups. One possible next step would be to combine all the features that are highly predictive of response accuracy into one model. However, the features are highly inter-correlated, so their joint performance does not necessarily exceed any single performance. Recall that for CREE, the TEConfidence feature alone outperforms the full model by two points F-score. To quantify the amount of inter-correlation, an average 67% of the variation in the UnigramToken feature can be explained by a linear combination of the other feature groups (*excluding* the NGram features) across the corpora. This explains the high UnigramToken baseline - the other features strongly co-vary with the NGram features and contribute relatively little additional information.

In order to find the most predictive feature combinations, we choose an extrinsically-motivated model-building strategy. We propose adding features in the order of processing effort necessary to produce them, with the motivation that any information that can be gained by simple means should not be duplicated by more costly methods. Starting from the NGram feature set, we incrementally add more features and monitor the performance to find the cut-off point at which the complete model performance has been approximated (or even out-performed).

Table 5 shows the results. Note that the NGram feature group as a whole often outperforms the UnigramToken baseline, since the higher-order NGram features in the feature group contribute additional information. Adding the TE features in the final step results in the full model performance. The intermediary results in bold face represent substantial increases in model performance. Underlined results numerically outperform the full model.

As expected after our discussion of feature patterns in Section 6, we find that for all corpora, subsets of the feature groups suffice to approximate full model performance. In four out of five cases, we even optimise performance by using fewer features.

There are few surprises in the feature groups that contribute substantially to model performance: Again, we see a strong reliance on the NGram and similarity features. For three out of the five corpora, the full model performance can be reached or exceeded just by these two feature groups. Adding dependency features further improves four out of five models (although the CSSAG improvement is negligibly small).

Feature Group	SEB	Beetle	CSSAG	CREG	CREE
UnigramToken Bsl	61.7	70.9	67.4	78.6	66.5
NGrams	**62.1**	**72.6**	**66.5**	**80.5**	**60.9**
+ Similarity	**62.6**	72.6	**69.4**	**82.5**	**67.4**
+ Dependency	**64.7**	70.9	69.5	**83.5**	**69.8**
+ Semantics	64.7	**73.4**	66.4	83.4	**70.7**
+ TE (full model)	**66.0**	72.6	66.6	81.5	70.2

Table 5: Performance in F-score when adding feature groups in order of processing effort. Boldfaced figures indicate a substantial contribution to model performance, underlined figures exceed full model performance.

SEB, Beetle and CREE profit from features from deep processing (semantics and TE). This matches our analysis above that the more varied language in content assessment corpora (and the highly-proficient paraphrasing that creates the reference answers from the reading text for CREE) can be successfully addressed by more abstract features from deeper levels of linguistic analysis.

For the individual corpora, there is a clear correspondence between feature groups with highly predictive features in Table 4 and useful feature groups in Table 5. Any feature group containing a feature that approximates full model performance within about four points F-score proves useful in incremental model construction. Interesting exceptions to this rule are CSSAG and CREE, where the semantic and TE features (CSSAG) or just the TE features (CREE) are individually predictive within four points F-score of the full model, but do not improve combined model performance. Since these features are added last, the information they contain appears to be already covered by the combination of the other feature groups. However, the best incremental CREE model still does not outperform the model only using TEConfidence (F=72.4). The CREG and SEB models profit from adding feature groups that alone are not extremely predictive (CREG: similarity and dependency features, SEB: similarity, dependency and TE features). These feature groups clearly add relevant new information given the backbone of NGram features.

In sum, we again find that the NGram and text similarity features are very predictive of response accuracy for all corpora. This is mirrored in the literature in the SemEval-2013 performance of the CU model (Okoye et al., 2013) that focuses on these feature types, or the strong results recently presented by Sultan et al. (2016), who use lexical overlap and vector-based text similarity features. Dependency features are also worth computing, as they further improve performance for four out of five models. Features from deeper linguistic processing levels are useful if the student answers differ from the reference answer by proficient paraphrasing (as opposed to insertions, deletions and re-orderings). This is the case whenever proficient speakers answer content assessment questions (or adapt the reference answer, as for the language skill assessment in CREE).

8 Conclusions

The goal of this paper was to identify highly predictive features for the short-answer grading task. We used five corpora from the content and language-skill assessment domains to ensure that our findings would generalise and verified that our full feature set approximates literature results.

The analyses found generally applicable features in the realm of shallow (Unigram) to medium (text similarity and dependency) linguistic analysis. Features on deeper processing levels were found to co-vary substantially with the shallow features. This explains why the lexical baseline is hard to break. Deeper features (semantic representations and TE) are however useful to model the higher levels of linguistic variation in our content-assessment corpora (as opposed to the language-skill corpora). There was no language-specific pattern to feature predictiveness.

These results serve as a starting point for future research into automated short-answer grading. Depending on the corpus type at hand, our feature recommendations can be used to quickly build a well-motivated basis model to expand by further deep or shallow features, according to corpus type.

Acknowledgements

The author would like to extend thanks to Michael Hahn for providing access to the LRS parser, to Valeriya Ivanova and Verena Meyer for their work on the CSSAG corpus and processing software and to Sebastian Padó for helpful discussions.

References

Stacy Bailey and Detmar Meurers. 2008. Diagnosing meaning errors in short answers to reading comprehension questions. In *Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications (BEA-3) at ACL'08.*, pages 107–115.

Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *Int J Artif Intell Educ*, 25:60–117.

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment. In *Proceedings of SemEval-2013*, pages 263–274.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Michael Hahn and Detmar Meurers. 2012. Evaluating the meaning of answers to reading comprehension questions: A semantics-based approach. In *The 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pages 326–336.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1).

Michael Heilmann and Nitin Madnani. 2013. ETS: Domain adaptation and stacking for short answer scoring. In *Proceedings of SemEval-2013*, pages 275–279.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2013. Softcardinality: Hierarchical text overlap for student response analysis. In *Proceedings of SemEval-2013*.

Bernardo Magnini, Roberto Zanoli, Ido Dagan, Katrin Eichler, Günter Neumann, Tae-Gil Noh, Sebastian Pado, Asher Stern, and Omer Levy. 2014. The Excitement Open Platform for textual inferences. In *Proceedings of the ACL demo session*.

Detmar Meurers, Ramon Ziai, Niels Ott, and Stacey Bailey. 2011a. Integrating parallel analysis modules to evaluate the meaning of answers to reading comprehension questions. *Special Issue on Free-text Automatic Evaluation. International Journal of Continuing Engineering Education and Life-Long Learning (IJCEELL)*, 21(4):355–369.

Detmar Meurers, Ramon Ziai, Niels Ott, and Janina Kopp. 2011b. Evaluating answers to reading comprehension questions in context: Results for german and the role of information structure. In *Proceedings of the TextInfer 2011 Workshop on Textual Entailment*, pages 1–9, Edinburgh, Scottland, UK. Association for Computational Linguistics.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 752–762. ACL.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *IWPT 03*, pages 149–160.

Ifeyinwa Okoye, Steven Bethard, and Tamara Sumner. 2013. CU: Computational assessment of short free text answers - a tool for evaluating students' understanding. In *Proceedings of SemEval-2013*, pages 603–607.

Niels Ott, Ramon Ziai, and Detmar Meurers. 2012. Creation and analysis of a reading comprehension exercise corpus: Towards evaluating meaning in context. In Thomas Schmidt and Kai Wörner, editors, *Multilingual Corpora and Multilingual Corpus Analysis*, Hamburg Studies in Multilingualism (HSM), pages 47–69. Benjamins, Amsterdam.

Niels Ott, Ramon Ziai, Michael Hahn, and Detmar Meurers. 2013. CoMeT: Integrating different levels of linguistic meaning assessment. In *Proceedings of SemEval-2013*, pages 608–616.

Ulrike Padó and Cornelia Kiefer. 2015. Short answer grading: When sorting helps and when it doesn't. In *Proceedings of the Nodalida-2015 workshop*.

Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. 2015. Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–106.

Frank Richter and Manfred Sailer. 2004. Basic concepts of lexical resource semantics. In Arnold Beckmann and Norbert Preining, editors, *European Summer School in Logic, Language and Information 2003. Course Material I, volume 5 of Collegium Logicum*, pages 87–143. Publication Series of the Kurt Gödel Society, Vienna.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*.

Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In *Proceedings of NAACL-HLT 2016*, pages 1070–1075.

Michael J. Wise. 1996. YAP3: Improved detection of similarities in computer program and other texts. In *SIGCSEB: SIGCSE Bulletin (ACM Special Interest Group on Computer Science Education)*, pages 130–134. ACM Press.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of COLING 2000*, pages 947–953.

Torsten Zesch, Omer Levy, Iryna Gurevych, and Ido Dagan. 2013. UKP-BIU: Similarity and entailment metrics for student response analysis. In *Proceedings of SemEval-2013*.