ETS®

Listening. Learning. Leading.®

Research Report
ETS RR–15-23

Automated Analysis of Text in Graduate School Recommendations

Michael Heilman

F. Jay Breyer

Frank Williams

David Klieger

Michael Flor

December 2015

ETS Research Report Series

Since its 1947 founding, ETS has conducted and disseminated scientific research to support its products and services, and to advance the measurement and education fields. In keeping with these goals, ETS is committed to making its research freely available to the professional community and to the general public. Published accounts of ETS research, including papers in the ETS Research Report series, undergo a formal peer-review process by ETS staff to ensure that they meet established scientific and professional standards. All such ETS-conducted peer reviews are in addition to any reviews that outside organizations may provide as part of their own publication processes. Peer review notwithstanding, the positions expressed in the ETS Research Report series and other published accounts of ETS research are those of the authors and not necessarily those of the Officers and Trustees of Educational Testing Service.

The Daniel Eignor Editorship is named in honor of Dr. Daniel R. Eignor, who from 2001 until 2011 served the Research and Development division as Editor for the ETS Research Report series. The Eignor Editorship has been created to recognize the pivotal leadership role that Dr. Eignor played in the research publication process at ETS.

ETS Research Report Series ISSN 2330-8516

RESEARCH REPORT

Automated Analysis of Text in Graduate School Recommendations

Michael Heilman, F. Jay Breyer, Frank Williams, David Klieger, & Michael Flor

Educational Testing Service, Princeton, NJ

Graduate school recommendations are an important part of admissions in higher education, and natural language processing may be able to provide objective and consistent analyses of recommendation texts to complement readings by faculty and admissions staff. However, these sorts of high-stakes, personal recommendations are different from the product and service reviews studied in much of the research on sentiment analysis. In this report, we develop an approach for analyzing recommendations and evaluate the approach on four tasks: (a) identifying which sentences are actually about the student, (b) measuring specificity, (c) measuring sentiment, and (d) predicting recommender ratings. We find substantial agreement with human annotations and analyze the effects of different types of features.

Keywords letters of recommendation; natural language processing; automated scoring

doi:10.1002/ets2.12070

Recommendations of people are very different from recommendations of commercial products and services, the focus of most natural language processing (NLP) research on recommendations (Blitzer, Dredze, & Pereira, 2007; Pang & Lee, 2004; Titov & McDonald, 2008). In low-stakes, impersonal recommendations of commercial products, many reviewers do not hesitate to use highly expressive language, even for negative reviews (e.g., this movie was horrifyingly terrible).

Here, we explore a recommendation scenario of a different sort, one in which the subject of the recommendation is often personally well known to the recommender (and vice versa) and where the stakes can be quite high. Specifically, we studied recommendations for graduate or professional schools. Unlike typical product reviewing scenarios, recommenders of prospective graduate students are typically more reluctant to use negative language about an applicant. Instead, they often either decline to provide a recommendation (which are never observed) or use lukewarm, vague, and generic language that avoids commitment (e.g., however, she could show a little more initiative).

We identified four challenges of analyzing graduate school recommendations:

- Much of the text in recommendations may express information other than evaluations of the applicant (e.g., information about the recommender, discussion of the evaluation process).
- The text about the applicant may be generic, lacking specific details or examples about the applicant to indicate that he or she is well known to the recommender.
- The text about the applicant may be either positive, negative, or neutral (e.g., negative text is likely to be associated with less qualified applicants).
- If numeric ratings accompany recommendation texts, as they do in our data, then the ratings for any individual recommendation form may not be consistent with its text (e.g., different recommenders may interpret rating scales differently). We may be able to provide more reliable scores by using models of the association between text and ratings learned from large samples of data.

In this work, we used NLP and machine learning techniques to automate these tasks, which are described in more detail in the Tasks section of this report. In doing so, we sought to answer the following research questions:

- How well do NLP models' predictions agree with human judgments, and how does this compare to human–human agreement?

Corresponding author: F. J. Breyer, E-mail: fbreyer@ets.org

- For which tasks does the approach we have developed outperform a bag-of-*n*-grams baseline?
- What words and phrases are associated with the scales for each task? For example, what words are associated with high recommender ratings?

By studying these questions, we make the following contributions to the literature: (a) a quantitative analysis of graduate school recommendations; (b) a demonstration of the potential for NLP to help analyze even high-stakes recommendation situations, assuming proper use (e.g., checks by humans); and (c) analyses of what types of linguistic features are predictive of text being about the student, specificity, and recommender ratings in graduate school recommendations. In addition, we open the door for future studies that directly predict indicators of graduate school success (e.g., acceptance and enrollment, grade point average) from recommendation texts, which could help admissions staff more accurately predict who will succeed in graduate school.

Data

In this section, we describe the data set of graduate school recommendations we used in our experiments as well as annotations we made on a subset of the data.

The original sample of recommendation forms consisted of 19,046 recommendation forms for 7,035 different graduate or professional school applicants. The forms were structured; recommenders were asked to evaluate applicants on six dimensions: (a) knowledge and creativity, (b) communication, (c) teamwork, (d) resilience, (e) planning and organization, and (f) ethics and integrity. For each dimension, the recommender was asked to rate the applicant with text statements related to the dimension (e.g., works well in group settings for teamwork) on four 5-point rating scales. The possible ratings were below average, average, above average, outstanding (top 5%), and truly exceptional (top 1%). A sixth choice, insufficient opportunity to evaluate, allowed recommenders to skip a prompt.[1]

It is worth noting that these recommendation texts, because they were collected as part of a structured form, may differ from traditional recommendation letters. For example, they may be more concise and topically focused on the targeted dimensions (e.g., knowledge and creativity) than a typical letter of recommendation would be. However, we believe that these data are generally representative of graduate school recommendations.

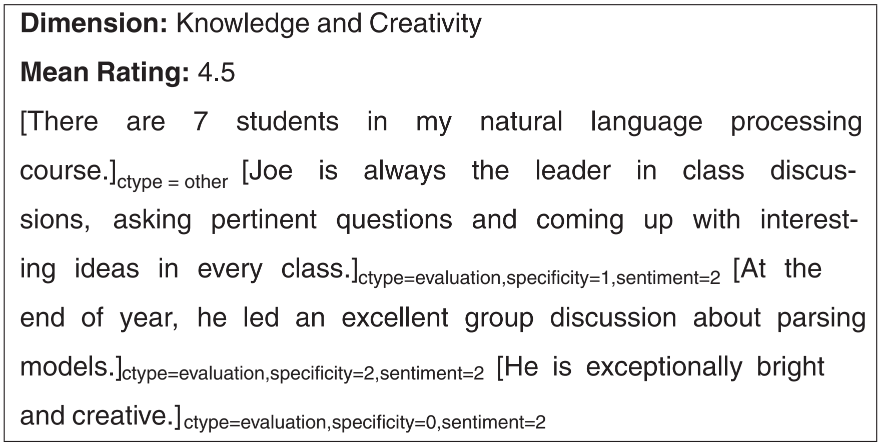
We selected a subset of the available nonblank text boxes for annotation by two annotators who worked independently after some training and practice. Part of this subset was doubly annotated to check interannotator agreement. For the set of doubly annotated texts, we selected one annotator's annotations and ignored the other's for training and evaluating the system. See the Data section of this report for counts of sentences and text boxes.[2]

We developed a taxonomy to annotate recommendation texts. First, annotators marked parts of the text that expressed evaluations of the applicant (these annotations correspond to the evaluation category for the content type task described in the Tasks section of this report). They also categorized the remaining text (e.g., for whether it was background information, comments about the form), but we collapsed these into a single *other* category because they were relatively infrequent and outside the scope of this research.

Second, for text marked as an evaluation of the applicant, the annotators marked which parts were generic content, specific content without examples, and specific content with examples. They also made annotations for sentiment polarity (very negative, negative, neutral, positive, and very positive).

Annotators were allowed to annotate texts by highlighting spans, and so the original annotations were created at the character level. For this work, however, we aggregated annotations to be at the sentence level, as discussed in the Tasks section of this report.

We initially split the data (by form) into two subsets: one for building the system (approximately two-thirds of the entire data set) and one for a held-out *test set* (the remainder). We then further split, by applicant, the first subset into a *training set* (approximately two-thirds) and a *development set* for evaluations during development (the remainder). Later, before running the experiments described here, we removed from the test set all the forms for applicants who appeared in the training or development sets, and we moved these forms into the training and development sets to avoid overlapping sets of applicants. The training set had 3,516 sentences from 1,020 nonblank text boxes, the development set had 1,883 sentences from 570 nonblank text boxes, and the test set had 1,385 sentences from 461 nonblank text boxes.



Dimension: Knowledge and Creativity

Mean Rating: 4.5

[There are 7 students in my natural language processing course.]$_{\text{ctype = other}}$ [Joe is always the leader in class discussions, asking pertinent questions and coming up with interesting ideas in every class.]$_{\text{ctype=evaluation,specificity=1,sentiment=2}}$ [At the end of year, he led an excellent group discussion about parsing models.]$_{\text{ctype=evaluation,specificity=2,sentiment=2}}$ [He is exceptionally bright and creative.]$_{\text{ctype=evaluation,specificity=0,sentiment=2}}$

Figure 1 An example, loosely adapted from our data set, to illustrate the content type (ctype), specificity, sentiment, and ratings tasks.

Finally, we created an *unannotated set* for the ratings task described in the Tasks section of this report from forms for applicants who did not appear in the other annotated sets. The unannotated set had 122,627 sentences from 39,882 nonblank text boxes.

Tasks

In this section, we describe the three prediction tasks we explored and how we adapted the data described in the Data section of this report for these tasks. An example, very loosely adapted from our data, is shown in Figure 1.

Content Type

The first task was to classify a sentence as to whether it expresses an evaluation of the student or only other information (e.g., background information about the recommendation writer). For this task, each text box was automatically split into sentences.[3] Each sentence was then assigned the label *evaluation* if the annotator marked any part of it as an evaluation of the applicant or *other* if not. The distribution of labels is skewed approximately 90% of the training set examples are labeled *evaluation*.

To measure performance on this task, we used (unweighted) kappa (Cohen, 1968), which corrects for chance agreement. We also reported F_1 scores for the evaluation class.

On the sample of texts that was doubly annotated, human–human agreement was $\kappa = .674$ and $F_1 = .966$.

Specificity

The second task was to identify the amount of specific information about the student in a sentence. This task depends on the content type task, and so we only included sentences whose gold standard label for the content type task was *evaluation*. Each sentence was given one of the following labels based on the annotations described in the Data section of this report:

- 0: a part of the sentence was annotated as a generic evaluation, but no specific or example annotations were present.
- 1: a part of the sentence was annotated as a specific evaluation, but no example annotations were present.
- 2: a part of the sentence was annotated as an example.

We posited an ordinal scale for these labels, where sentences labeled 0 provide the least specific information about an applicant and those labeled 2 provide the most specific information. The distribution of labels is skewed: Approximately 70% of the training set examples are labeled 0, 21% are labeled 1, and 9% are labeled 2.

To measure performance on this task, we used quadratic-weighted kappa (hereinafter, κ_{quad}) and unweighted kappa (Cohen, 1968). Quadratic-weighted kappa is more suitable for ordinal data, whereas unweighted kappa is more suitable for categorical data. Both are commonly used in educational measurement and psychology. We used the quadratic-weighted kappa as the primary evaluation metric (e.g., for tuning and significance testing).

On the sample of texts that was doubly annotated, human–human agreement was $\kappa_{\text{quad}} = .628$ and $\kappa = .457$.

Note that whereas we have learned a model of the specificity of single sentences, one could aggregate sentence-level predictions from this model to estimate the specificity of a complete recommendation (e.g., by summing, averaging).

Sentiment

The third task was to identify whether a sentence evaluating the student expresses positive, negative, or neutral sentiment. As for the specificity task, we only included sentences whose gold standard label for the content type task was evaluation. Each sentence was given one of the following ordinal labels based on the annotations described in the Data section of this report: 0 if any negative or very negative annotations were present (i.e., marked part or all of the sentence); 2 if any positive or very positive annotations (but no negative or very negative annotations) were present; and otherwise 1 (for sentences only marked neutral). We collapsed the original scale because doing so led to higher interannotator agreement, probably owing to the distinction between the positive and very positive (or negative and very negative) categories not being clear. The distribution of labels is very skewed: Approximately 1% of the training set examples are labeled 0 (i.e., negative), 1% are labeled 1 (i.e., neutral), and 98% are labeled 2 (i.e., positive).

We evaluated performance with the Pearson's correlation coefficient between the system outputs and the gold standard values (both unrounded). We also report Kendall's tau.[4]

On the sample of texts that was doubly annotated, human–human agreement on the 3-point scale was $r = .627$ and $\tau = .453$.[5]

Ratings

The fourth task was to predict the ratings provided by recommenders from the text that they entered. Thus the input for this task was the entire text in a text box (i.e., multiple sentences) rather than a single sentence, unlike for the content type and specificity tasks.

For simplicity, we converted the ratings from the original ordinal categories (above average, average, etc.) into 1–5 values. We then assigned the mean of the four ratings to be the label of each text. Recall from the Data section of this report that four 5-point rating prompts accompanied the dimension-specific text box. For cases where recommenders did not provide ratings for all prompts, the mean of the ratings they provided (i.e., the mean of up to three rating values instead of four) was used instead. Text boxes with no accompanying ratings were not used for this task (94 for the unannotated set, 0 for the development set, and 1 for the test set).

For this task, because we did not need annotations, we trained models on the unannotated set (see the Data section of this report). The mean ratings (i.e., the labels) tended to be quite high, with a mean of 4.34 ($SD = .65$) in the unannotated set.

As for the sentiment task, we used Pearson's correlation coefficient between the system outputs and the gold standard values (both unrounded). We also reported Kendall's tau.

Note that it was not straightforward to estimate human–human agreement for this task because the ratings are very subjective (e.g., two recommenders may have very different relationships to the applicant) and because we did not have multiple recommendations for all applicants.

Methods

In this section, we described our approach to the preceding tasks, including the models and features we used.

Statistical Models

We used ℓ_2-regularized generalized linear models for all three tasks.[6] For the binary content type task, we used ℓ_2-regularized logistic regression. We assigned class weights inversely proportional to class frequencies to avoid a model that overpredicts the more frequent evaluation category.

For the ordinal specificity, sentiment, and ratings tasks, we used ridge regression (i.e., ℓ_2-regularized least squares regression).[7]

For ridge regression, after training an initial model, we applied the following linear transformation to the predictions so that they better matched the gold standard distribution. Vector $\mathbf{y}_{\text{train}}$ was the vector of the training set gold standard labels, $\hat{\mathbf{y}}_{\text{train}}$ was the vector of predictions for the training set from the initial model, M returned the mean of a vector, and SD returned the standard deviation:

$$\hat{y}_{\text{new}} = \frac{\hat{y} - M\left(\hat{\mathbf{y}}_{\text{train}}\right)}{SD\left(\hat{\mathbf{y}}_{\text{train}}\right)} \times SD\left(\mathbf{y}_{\text{train}}\right) + M\left(\mathbf{y}_{\text{train}}\right). \tag{1}$$

Note that this transformation was performed on the training set.

To select hyperparameter values, we used fivefold cross-validation grid search on the training data, with all of the sentences (or text box entries) from an applicant contained within a single fold (i.e., rather than being spread across multiple folds). The grid had a single dimension: The values of 10^q for $q \in \{-4, -3, \ldots, 4\}$ were explored for regularization hyperparameter q.[8] For the grid search scoring function, we used kappa for the content type task, quadratic-weighted kappa for the specificity task, and Pearson's correlation coefficient for the sentiment and ratings tasks.

Features

The models for all four tasks used the same set of features, with minor variations to account for the fact that the content type, specificity, and sentiment tasks are at the level of sentences, while the ratings task was at the level of text boxes.

In addition to the features described later, an intercept, or bias, feature *BIAS* with a constant value of 10 is included so that the other features were not used to model which labels were most likely in general. Using 10 rather than 1 lessened the effect of regularization on the intercept.[9] We developed this feature set through preliminary evaluations on the development set.[10]

Length Features

The models included binary features *NTOKS_GRx* indicating whether the number of tokens in the sentence (or text, for the ratings task) exceeds x, where $x \in \{2, 4, 8, 16, 32, 64\}$.

Part-of-Speech Features

The models included, for each possible part-of-speech (POS) tag, a binary feature with the prefix *POS*, indicating the presence of that POS in the sentence (or text).[11]

Word n-Gram *Features*

The models included, for each word n-gram in the training set (for $n \in \{1, 2, 3\}$), a binary feature indicating the presence of that n-gram in the sentence (or text). These n-grams were computed from a preprocessed (see the Features subsection of this report) version of the original text.

Sentiment Features

The models included binary features indicating the presence of at least one positive, negative, or neutral word in the sentence (or text), after the preprocessing described in the Features subsection of this report. The labels were, for example, *POSITIVE*.

To compute these features, we used the sentiment lexicon from Beigman Klebanov, Madnani, and Burstein (2013). Each word in the lexicon has positive values between 0 and 1 for positivity, negativity, and neutrality, and these three values sum to 1. These values were estimated through crowdsourcing. For computing features in this model, we defined positive words to be those with a positivity value greater than .9. Negative words were those with a negativity greater than .9. Other words in the lexicon were considered neutral. Words that were not in the lexicon were not considered positive, negative, or neutral.

Surrounding Sentence Features

For the content type and specificity tasks, which were at the sentence level, the models included separate instances of the POS, word *n*-gram, and sentiment features for the previous sentence as well as the next sentence (e.g., in addition to a *POS* VBD POS feature for a past tense verb in the current sentence, there were also *PREV* *POS* VBD and *NEXT* *POS* VBD features). No such features were included for the text-level ratings task.

Frequency Features

The model included three features to model how frequent the words in the sentence (or text) were in a larger sample of English text—specifically, *New York Times* (NYT) stories from 2008 in the fourth edition of the gigaword corpus (Parker, Graff, Kong, Chen, & Maeda, 2009). We precomputed counts of all words in the NYT and filtered out words occurring fewer than 100 times to produce the final vocabulary.

The features were as follows, where $c(w_i)$ was the number of times that a word token i in the input sentence (or text) appeared in the NYT sample; N was the number of words in the input; $oov(w_i)$ is 1 if the word was not in the NYT vocabulary, and 0 otherwise; and log was the natural logarithm function. Note that $c(w_i) = 0$ for words not in the NYT vocabulary:

$$\frac{1}{N}\sum_{i=1}^{N} c\left(w_i\right),$$

$$\frac{1}{N}\sum_{i=1}^{N} \log\left(c\left(w_i\right) + 1\right),$$

$$\frac{1}{N}\sum_{i=1}^{N} oov\left(w_i\right).$$

The computation of these features was based on preprocessed (see the Features subsection of this report) versions of the input and NYT texts.

Token Preprocessing

An important issue in handling these high-stakes recommendations was that we wanted to avoid learning spurious associations that might unfairly affect future applicants. For example, we did not want the model to predict that a recommendation was negative simply because several people with the same first name received negative recommendations in the training set. Therefore, using metadata provided with the input text, the model performed the following preprocessing steps on each token. These steps generalized the text to reduce personal information (and also to simplify punctuation and numbers):

- If the token matched the applicant's first name, it was replaced by *FIRST_NAME*.
- Otherwise, if the token was a nickname of the applicant's first name, it was replaced by *NICK_NAME*. For this step, we used the database of nicknames created by Carvalho, Kiran, and Borthwick (2012).
- Otherwise, if the token matched the applicant's last name, it was replaced by *LAST_NAME*.
- Otherwise, if the token was he, she, him, her, his, her or hers, himself, or herself, it was replaced by *HE_SHE*.
- Otherwise, if the lowercased token was miss, mr., ms., or mrs. (or a version of those without the period), it is replaced by *MR_ETC*.
- Otherwise, if the token contains a numeric digit, it was replaced by *NUMBER*.
- Otherwise, if the token contained no letters or numbers, it was replaced by *PUNCT*.
- Otherwise, if the token was not a stopword[12] and had a capital letter, it was replaced by *CAP* (even when at the beginning of a sentence). This aggressive preprocessing step helped the model avoid variations of names not captured in the preceding steps, including misspellings and less frequent nicknames.

Table 1 Development Set Performance of Models Excluding Each of the Various Types of Features From the Full Model

	Content type		Specificity		Sentiment		Ratings	
	κ	F_1	κ_{quad}	κ	r	τ	r	τ
Full model	0.474	0.961	0.516	0.320	0.187	0.105	0.327	0.251
– Length	0.452[a]	0.959	0.504[a]	0.308	0.186	0.105	0.323	0.247
– POS	0.472	0.963	0.465[a]	0.284	0.172	0.087	0.317	0.241
– *n*-grams	0.212[a]	0.855	0.468[a]	0.274	0.111[a]	0.099	0.168[a]	0.146
– Sentiment	0.402[a]	0.956	0.509	0.312	0.163[a]	0.085	0.321	0.246
– Surrounding	0.388[a]	0.943	0.493	0.299	0.208	0.105	–	–
– Frequencies	0.476	0.961	0.510	0.307	0.186	0.101	0.326	0.249
Bag-of-*n*-grams	0.375[a]	0.944	0.454[a]	0.278	0.204	0.092	0.320	0.244

Note. −*X* denotes a model with all types of features except *X*. For example, −*length* indicates a model with all but the length features. Note that the surrounding feature type is not relevant for the ratings task because it pertains to text boxes rather than single sentences. POS = part of speech.

[a] Indicates that the task-specific primary evaluation metric value for a system variation was significantly different from the full system. Confidence intervals were only computed for the primary metrics (see text for details). For each task, the primary metric is in the leftmost column.

Experiments

In this section, we describe experiments to evaluate our models for the three tasks.

Bag-of-*n*-Grams Baseline

In our experiments, we compared our system to a relatively simple bag-of-*n*-grams approach, a common baseline for text analysis tasks such as these. To implement this, we removed all features except for the bias feature and the word *n*-gram features for the current sentence (or text, for the ratings task).

Feature Ablation

We performed feature ablation experiments to study the effects of the different types of features described in the Statistical Models subsection of this report. For each task, we trained a model with all the features and then models lacking each feature type. We estimated parameters with the annotated training set for the content type, specificity, and sentiment tasks and with the larger unannotated set for the ratings task (see the Data section of this report). We then computed performance on the development set. The results are shown in Table 1.[13]

For each variation of the full system, we computed a 95% BC_a bootstrap (Efron & Tibshirani, 1993) confidence interval, with 50,000 replications, for the difference in performance between the performance of the full model and the performance of the variation (e.g., the model without length features). A confidence interval that does not include zero indicates that the difference is not likely to be because of chance. We computed these confidence intervals only for the primary evaluation metric for each task: unweighted kappa for specificity, quadratic-weighted kappa for specificity, and Pearson's correlation coefficient for sentiment and ratings (see the Tasks section of this report). The results are shown in Table 1.

It is interesting to look at the feature types that led to the largest drops in performance for each task, because those may be the most important. For example, for the content type, sentiment, and ratings tasks, word *n*-gram features appear to play a very important role because the models without *n*-gram features performed much worse than the full models. For example, for the content type task, performance was $\kappa = .474$ for the full model compared with $\kappa = .212$ for the model without *n*-gram features. In contrast, for the specificity task, word *n*-grams do not appear to be the most important: for specificity, removing POS features led to the largest drop in performance, from $\kappa_{\text{quad}} = .516$ to $\kappa_{\text{quad}} = .465$.

Including *n*-grams, POS, and sentiment features of the surrounding sentences appears to help for the content type and specificity tasks, because not doing so (– surrounding) led to drops in performance, though the drop for the specificity task was not statistically significant. In future work, it might be worth addressing these tasks with sequence models over sentences, such as conditional random field models (Lafferty, McCallum, & Pereira, 2011), rather than with the single-sentence models used here.

Table 2 Test Set Performance of the Full Model and Bag-of-*n*-Grams Baseline Model for Each of the Four Tasks

	Content type		Specificity		Sentiment		Ratings	
	κ	F_1	κ_{quad}	κ	r	τ	r	τ
Full model	0.467	0.952	0.516	0.279	0.272	0.119	0.356	0.251
Bag of *n*-grams	0.428	0.939	0.457[a]	0.260	0.343[a]	0.164	0.331[a]	0.228

[a] Indicates that the task-specific primary evaluation metric value for the baseline was significantly different from the value for the full system. Confidence intervals were only computed for the primary metrics (see text for details). For each task, the primary metric is in the leftmost column.

Test Set Performance

In this section, we present results on the held-out test set (see the Data section of this report). We trained the content type, specificity, and sentiment models on the combination of the annotated training and development sets. We trained the ratings model on just the large unannotated set, as before. The results are shown in Table 2.

While the full models appear to have captured much of the relevant information for each task, human–machine agreement was still somewhat lower than human–human agreement. For the content type task, test set human–machine agreement was $\kappa = .467$ compared with the $\kappa = .674$ human–human agreement on the doubly annotated set (which partially overlaps with the test set, as explained in the Data section of this report). For the specificity task, human–machine agreement was $\kappa_{\text{quad}} = .516$ compared with $\kappa_{\text{quad}} = .628$ human–human agreement. For the sentiment task, human–machine agreement was $r = .272$ compared with $r = .627$ human–human agreement.

We observe that the relatively low performance of the model for the ratings task may be related to a lack of consistency in the ratings themselves: previous work has found that recommendations about the *same* applicant from *different* faculty members are less correlated than recommendations from the *same* faculty member about *different* applicants (Aamodt, Bryan, & Whitcomb, 1993; Baxter, Brock, Hill, & Rozelle, 1981).

The relatively low performance of the sentiment model is likely because of a paucity of training data for the negative and neutral classes. Each of these constituted only about 1% of the training data because almost all of what is said about an applicant in most recommendations is positive (and negative things are apparently left unsaid).

The full model with all features outperformed the bag-of-*n*-grams baseline for the content type, specificity, and ratings tasks, though the differences were small for the content type and ratings tasks. The full model performed worse than the baseline for the sentiment task.

To check whether these differences are likely to be because of random chance, we computed 95% BC_a bootstrap (Efron & Tibshirani, 1993) confidence intervals from 50,000 replications for the differences in performance for each task, as in the Feature Ablation subsection of this report. Cases where the interval for the difference did not include 0, thus indicating a significant difference, are indicated in Table 2. Note that the tests were computed only for the primary evaluation metric for each task.

Interestingly, for the content type task, the performance of the full model was statistically significantly different from the baseline on the development set (in the Feature Ablation subsection of this report) but not the test set. Conversely, for the sentiment and ratings tasks, the full model was statistically significantly different from the baseline on the test set but not the development set. We speculate that this could be because of the relatively small sample size for the annotated data sets used for evaluation. For the content type task, another possibility is that the baseline bag-of-*n*-grams model depends more on the availability of large amounts of annotated data, and so it benefited more than the full model from having access to the combined training and development sets for the test set experiments.

Model Exploration

The most strongly weighted features, according to absolute values, for the full models from the Test Set Performance subsection of this report for each task are shown in Table 3.

The model for the content type task is fairly straightforward. Features such as third person pronouns and indicators for the name of the student had strong positive weights, whereas features such as first person pronouns (e.g., for when a recommender talks about himself or herself instead of the applicant) had strong negative weights.

Table 3 Most Strongly Positively and Negatively Weighted Features in the Full Model for Each Task

Content type		Specificity		Sentiment		Ratings	
Wgt.	Feature	Wgt.	Feature	Wgt.	Feature	Wgt.	Feature
1.107	*HE_SHE*	0.117	VBD	0.189	*BIAS*	0.397	*BIAS*
0.755	*POSITIVE*	0.075	*PREV* VBD	0.041	*NTOKS_GR2*	0.105	Exceptional
0.523	*FIRST_NAME*	0.064	*NEXT* VBD	0.031	If	0.085	Integrity
0.441	*NEXT* *HE_SHE*	0.052	The	0.026	And	0.076	Ethical
0.394	VBZ	0.050	*NTOKS_GR16*	0.025	Not only	0.072	Truly
0.339	*PUNCT* *HE_SHE*	0.045	*NTOKS_GR32*	0.021	*NEXT* never	0.068	Extremely
0.335	RB	0.045	*CAP* *CAP*	0.021	*PREV* do	0.065	Exceptionally
0.297	And	0.041	VBG	0.021	Will	0.064	Never
0.265	Well	0.041	*CAP*	0.021	Do	0.063	Trust
0.265	*PREV* NN	0.040	This	0.020	*HE_SHE*	0.059	Not only
…	…	…	…	…	…	…	…
−0.286	*NEXT* *FIRST_NAME*	−0.026	Never	−0.061	*CAP* to	−0.042	Written
−0.307	At	−0.027	*NEXT* *HE_SHE*	−0.061	*NEXT* down	−0.042	*NTOKS_GR8*
−0.308	As	−0.030	*PREV* *POSITIVE*	−0.062	Sometimes	−0.043	But
−0.319	My	−0.035	*POSITIVE*	−0.064	To be more	−0.044	Little
−0.319	*CAP* *PUNCT*	−0.037	i	−0.065	Bit	−0.046	Sometimes
−0.349	Have	−0.039	*PREV* VBZ	−0.070	Too	−0.047	Language
−0.353	We	−0.040	Always	−0.071	A bit	−0.048	Questions
−0.398	*CAP*	−0.040	Is	−0.071	More	−0.065	Above average
−0.400	I	−0.041	*HE_SHE*	−0.071	But	−0.070	Good
−0.402	VBP	−0.066	VBZ	−0.079	Too much	−0.124	Average

Note. The features and their labels are described in the Statistical Models subsection of this report.

The specificity model seemed to rely heavily on POS. For example, its most strongly positive feature was whether the input sentence contained a past tense verb (*POS* VBD), probably because the past tense is typically used when providing detailed examples.

The model for the sentiment task similarly relied heavily on lexical features. Because the positive class was so predominant (about 98% of training set examples), very common words and phrases had positive weights, making them behave almost like extra bias (i.e., intercept) features. The negatively weighted sentiment task features are more interesting: here, a number of words and phrases expressed reservation (e.g., sometimes, a bit).

The model for predicting recommenders' ratings directly appeared to rely almost exclusively on word unigram features. It may be the case that the more general features (e.g., POS) are less useful for this task because they are just binary indicators. Using count features instead of binary features, possibly weighted with tf-idf or some other transformation, might lead to improved performance for this task. Nonetheless, it is interesting to observe the strongly weighted words for this model. The positive words included a number of adjectives and adverbs that express unqualified praise (e.g., exceptional, truly, extremely). Conversely, the negatively weighted words were not the sort of unqualified negative adjectives, such as terrible, that one might see in product or service reviews. Instead, words and phrases like average, good, and above average received negative weights in the model. Also, many words that are typically used to express reservations and qualifications—that is, potential hedge cues—received negative weights (e.g., but, sometimes, little).

Related Work

As part of this work, we classified sentences according to specificity. A closely related aim was distinguishing objective language from subjective language (Benamara, Chardon, Mathieu, & Popescu, 2011; Pang & Lee, 2004). Although subjectivity analysis can be used to filter out objective language (Pang & Lee, 2004), we believe that the objective language (e.g., in specific examples) is often the most informative for graduate school recommendations: recommenders who do not know an applicant well tend to use generic language, which may not be very informative, even though it is subjective. It would be interesting to compare predictions for subjectivity and specificity.

In addition, previous work on product reviews has predicted how helpful the reviews themselves are (Danescu-Niculescu-Mizil, Kossinets, Kleinberg, & Lee, 2009; Ghose & Ipeirotis, 2011), which is related to the aim of the specificity task. Whereas some product review data sets have ratings of helpfulness aggregating from hundreds of users (e.g., indicating that 35 of 50 people found a review helpful), such helpfulness data are not readily available for our data, and achieving interannotator agreement might be challenging.

Another closely related area of research is on detecting hedges — that is, language used to express uncertainty on behalf of the writer. Farkas, Vincze, Móra, Csirik, and Szarvas (2010) described a shared task on the topic. We observe substantial overlap between the features we used and those used by many of the submissions to the hedging task (e.g., words, POS; see the Statistical Models subsection of this report). Also, some of the words picked up as strongly weighted negative features in the ratings model could be considered hedge cues (in the Model Exploration section of this report).

Various research outside of the computational linguistics field has studied recommendations (Aamodt et al., 1993; Baxter et al., 1981; Vannelli, Kuncel, & Ones, 2007), though typically on a relatively small scale.

A more general connection can also be drawn to the large literature on automated scoring of student essays (Shermis & Burstein, 2013; Shermis & Hamner, 2012) and short answers (Dzikovska et al., 2013). In that area and in this work, the goal is to use NLP to support educational decision making.

Conclusion

In this report, we presented an NLP approach to analyzing graduate school recommendation texts. The same approach, which includes features for *n*-grams, POS, text length, sentiment words, and word frequencies, was used to train models for four tasks: identifying the parts of the text that were actually evaluations of the applicant, measuring the level of specificity in sentences evaluating the applicant, measuring whether the sentiment expressed in a sentence evaluating the applicant is positive or negative and predicting human recommender ratings from multisentence text box entries.

In response to the three research questions in this report, we found the following:

- Predictions from the NLP approach we presented in this report were found to agree with human judgments, but the agreement was somewhat lower than our estimate of human – human agreement.
- The approach we presented outperformed a simple bag-of-*n*-grams baseline on the test set for the specificity and ratings tasks but not for the content type or sentiment tasks.
- Different types of features were weighted highly for the three tasks (e.g., POS features for the specificity task and word *n*-grams for the ratings prediction task).

These results support various future works. In particular, NLP may help quantify the texts of recommendations by producing statistics about how much specific information or positive language is present. These statistics could be provided to academic institutions, or they could be integrated into a statistical model for predicting future academic or career success (e.g., from recommendations as well as grade point averages, standardized test scores, and other inputs).

Acknowledgments

The authors would like to thank those who reviewed drafts of this report, including Nitin Madnani, Aoife Cahill, and Beata Beigman Klebanov. This research was supported in part by allocation funding from the Center for Academic and Workforce Readiness and Success, Research and Development, ETS to the second author. All statements expressed in this article are the authors’ and do not reflect the official opinions or policies of the authors’ host affiliation.

Notes

1 For each dimension, recommenders could provide additional information in a text box. In addition, there was a space for overall comments, which we do not make use of in this work.

2 The doubly annotated subset overlapped with the training, development, and test sets described in the Data section of this report. For the doubly annotated set, the text boxes for the knowledge and creativity dimension were used as a pilot or practice run, for the annotation process. We treated the text boxes with these pilot annotations for the knowledge and creativity dimension as unannotated data (i.e., we did not use the annotations in our experiments).

3 We used proprietary tools for sentence splitting and word tokenization.

4 We used the τ_B implementation in scipy (http://www.scipy.org/), which performed an adjustment for ties.

5 Human – human agreement on the uncollapsed, 5-point sentiment scale was lower: $r = .461$ and $\tau = .341$.

6 We used the implementations in the scikit-learn package (Pedregosa et al., 2011, version 0.14.1).

7 We also tried logistic regression for the specificity task, thereby treating the data as categorical rather than ordinal. Performance was similar to what is shown in Table 1 ($\kappa = .324$ and $\kappa_{\text{quad}} = .488$), so we chose the ridge regression model, which has fewer parameters.

8 The regularization hyperparameters were C and α for the logistic and ridge regression implementations, respectively, in the scikit-learn package (Pedregosa et al., 2011). Large C or small α values encouraged smaller weights. Also, during the cross-validation grid search, the transformation in Equation 1 was applied on the training folds in each iteration not on the entire training set.

9 The actual labels, such as *BIAS*, were not important to the model, but we have included some of them here because they are mentioned later, in the Model Exploration of this report.

10 In preliminary experiments, we tried the domain adaptation technique from Daume (2007) to have dimension-specific features, but we did not observe general improvements in performance over models with just generic features.

11 We used the default POS tagger in NLTK (Bird, Klein, & Loper, 2009). POS tags are computed from the original tokens, not the preprocessed versions (the Features subsection of this report).

12 We used the list of stopwords from NLTK (Bird et al., 2009).

13 For computing confidence intervals in this section and in the Test Set Performance subsection of this report, we made the simplifying assumption that all data points (i.e., sentences or text boxes) are independent.

References

Aamodt, M. G., Bryan, D. A., & Whitcomb, A. J. (1993). Predicting performance with letters of recommendation. *Public Personnel Management*, *22*, 81–91.

Baxter, J. C., Brock, B., Hill, P. C., & Rozelle, R. M. (1981). Letters of recommendation: A question of value. *Journal of Applied Psychology*, *66*, 296–301.

Beigman Klebanov, B., Madnani, N., & Burstein, J. (2013). Using pivot-based paraphrasing and sentiment profiles to improve a subjectivity lexicon for essay data. *Transactions of the Association for Computational Linguistics*, *1*, 99–110.

Benamara, F., Chardon, B., Mathieu, Y., & Popescu, V. (2011, November). *Towards context-based subjectivity analysis*. Paper presented at the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Sebastopol, CA: O'Reilly Media.

Blitzer, J., Dredze, M., & Pereira, F. (2007, June). *Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification*. Paper presented at the 45th annual meeting of the Association of Computational Linguistics, Prague, Czech Republic.

Carvalho, V., Kiran, Y., & Borthwick, A. (2012). The intelius nickname collection: Quantitative analyses from billions of public records. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Retrieved from http://www.aclweb.org/anthology/N12-1075

Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*, 213–220.

Danescu-Niculescu-Mizil, C., Kossinets, G., Kleinberg, J., & Lee, L. (2009, April). *How opinions are received by online communities: A case study on amazon.com helpfulness votes*. Paper presented at the 18th international conference on the World Wide Web, New York, NY.

Daume III, H. (2007). *Frustratingly easy domain adaptation*. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P07-1033

Dzikovska, M. O., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., … Dang, H. T. (2013, June). *Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge*. Paper presented at *SEM 2013: The First Joint Conference on Lexical and Computational Semantics, Atlanta, GA.

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. Boca Raton, FL: Chapman and Hall/CRC Press.

Farkas, R., Vincze, V., Móra, G., Csirik, J., & Szarvas, G. (2010). The CONLL-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Retrieved from http://www.aclweb.org/anthology/W10-3001

Ghose, A., & Ipeirotis, P. G. (2011). Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Transactions on Knowledge and Data Engineering*, *23*, 1498–1512.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2011, June). *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. Paper presented at the 18th International Conference on Machine Learning, San Francisco, CA.

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting of the Association for Computational Linguistics (ACL-04)*. Retrieved from http://www.aclweb.org/anthology/P/P04/P04-1035.pdf

Parker, R., Graff, D., Kong, J., Chen, K., & Maeda, K. (2009). *English gigaword* (4th ed.). Philadelphia, PA: Linguistic Data Consortium.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825 – 2830.

Shermis, M. D., & Burstein, J. (2013). *Handbook of automated essay evaluation: Current applications and new directions*. New York, NY: Routledge Academic.

Shermis, M. D., & Hamner, B. (2012, April). *Contrasting state-of-the-art automated scoring of essays: Analysis*. Paper presented at the annual meeting of the National Council on Measurement in Education, Vancouver, BC.

Titov, I., & McDonald, R. (2008, June). *A joint model of text and aspect ratings for sentiment summarization*. Paper presented at ACL-08: HLT, Columbus, OH.

Vannelli, J., Kuncel, N. R., & Ones, D. S. (2007, April). A mixed recommendation for letters of recommendation. In N. R. Kuncel (Chair), *Alternative predictors of academic performance: The glass is half empty*. Symposium conducted at the annual meeting of the National Council on Measurement in Education, Chicago, IL.

Suggested citation:

Heilman, M., Breyer, F. J., Williams, F., Klieger, D., & Flor, M. (2015). *Automated analysis of text in graduate school recommendations* (ETS Research Report No. RR-15-23). Princeton, NJ: Educational Testing Service. http://dx.doi.org/10.1002/ets2.12070

Action Editor: Beata Beigman Klebanov

Reviewers: Jill Burstein and Aoife Cahill