Distinguishing different types of conference submissions: the ACL case study*

Hong Wang, Barbara Di Eugenio, Shuyang Lin, Clement T. Yu
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607, USA
{hwang207, bdieugen, slin38, cyu}@uic.edu

Abstract

Many conferences in AI and NLP call for long and short papers; and satellite workshops co-locate with the main conference. In this work, we focus on distinguishing full from short from workshop papers, as submitted to some recent ACL conferences. We propose a framework that takes into account both metadata and content of the paper. To extract metadata, we devised a full-fledged paper parser. SVM models outperform the only previously published results by at least 3.6% as concerns distinguishing full from workshop papers. Metadata (number of tables/formulas), syntactic feature (syntactic complexity) and term TF-IDF score distinguish full from short papers, whereas the topic also distinguishes full from workshop papers.

1 Introduction

When preparing a paper for a conference, its authors often wonder whether the work is better suited for a full or a short paper; or whether it should be submitted to one of the attendant workshops. Especially less experienced authors often refer to more experienced ones, such as their advisors, to help answer this question. This paper studies the following two problems: (1) Which features of papers, if any, correlate with different kinds of papers at ACL? (2) Assuming that full papers carry more prestige, to what extent can an automatic program assess whether a potential submission reaches full paper quality?

The first contribution of Automatic Assessor (ACL-AA), the system we propose, is a full-fledged paper parser that extracts both metadata and content from the paper. The second contribution is the features that range from metadata, such as the number of formulas or of tables, to paper content, to contextual features, such as the prominence of its authors, or the popularity of certain techniques. After generating the features, we use supervised learning (SVMs). Different models are built corresponding to different assessment tasks; we use 10-fold cross-validation to select the best combination of features and tune parameters for the corresponding models. Our evaluation on testing data set shows that the approach we propose always outperforms baseline or previously published results, whatever is available, by at least 18.9% when distinguishing full from short papers, and by more than 3.6% when distinguishing full from long workshop papers. We obtain better results than (Bergsma et al., 2012), the only previous paper we know of that addresses this task. Complexity of sentences, metadata like number of formulas per page and TF-IDF score of paper abstract terms are useful to differentiate full from short papers. When distinguishing full from workshop papers, the topic of the paper, together with its metadata, TF-IDF score and sentence complexity are the most predictive features, sometimes together with the prominence of authors and the popularity of certain techniques.

Motivations are two-fold. First, our work will be the core of an automatic reviewer system, similar in spirit to automatic graders, e.g. (Burstein et al., 2003). It would support both reviewers and authors, the former in assessing novelty, relevance etc. of papers, the latter in preparing better papers to start with. It could also support professional societies interested in investigating features of the review process, including potential bias (see our RANK feature). Second, computational stylometry is a novel area of research, which "aims to recover useful attributes of documents from the style of the writing" (Bergsma et al., 2012). It can sup-

*Manuscript for Chicago Colloquium on Digital Humanities and Computer Science (DHCS), 2014

port linguistic or sociological analysis of a body of literature, including insight into collaborations.

The potential applicability of our work does not simply concern helping authors decide which sort of paper to submit to ACL. Looking at the task from the point of view of reviewers, a sound automatic model could provide additional insight for borderline cases, or when reviewers disagree, especially as full papers are concerned. Additionally, some of the features that the model highlights as predictive are indicative of qualities that reviewers are called to comment about, such as novelty, relevance, thoroughness. Hence, these features may be useful to authors, especially younger researchers, in order to submit better papers to start with.

2 Related Work

(Bergsma et al., 2012) is the only work we know of that specifically focuses on distinguishing ACL conference from workshop papers (they also address whether the paper is written by a native or non-native speaker; and by a male or female). Apart from unigrams and bigrams, they use features derived from *style words* such as Latin abbreviations; and different types of syntactic features computed over all the parse trees derived from each document; the ones they find more effective, although more computationally intensive, are reranking features from (Charniak and Johnson, 2005). Their data is derived from the ACL Anthology Network (AAN)[1]: they train on papers from year 2001 to 2007, and test on papers from 2008 to 2009. Their best result on distinguishing ACL main session from workshop papers (F1=66.7%) is obtained when they use other NLP conferences as well (e.g. Coling, EMNLP) to train the model. We will show that on our "vs workshop" task we obtain F1=70.3% on distinguishing full from (long) workshop papers, and F1= 74.6% on distinguishing full from short papers (a task they do not engage in). Some of our features attempt at encoding novelty and relevance as well.

Other work explores features of papers to predict impact, such as number of citations and downloads. Mostly they use bag-of-words features, augmented with similarity measures (Bethard and Jurafsky, 2010) or simple metadata (Yogatama et al., 2011). Interestingly (Bethard and Jurafsky, 2010) employs topic similarity, computed via Latent Dirichlet Allocation (LDA) (Blei et al., 2003). We use LDA topic models as well. However (Bethard and Jurafsky, 2010) sets the number of topic to N=100, while we experiment with many different N's. Topic models are also used by an orthogonal line of work that assesses the content of essays, namely, automated essay scoring (AES) systems. The first attempts at automatic graders go back to the 1960s (Page, 1966), but use only surface features of the text. Modern systems like $Criterion^{SM}$ (Burstein et al., 2003) and AEA (Kakkonen et al., 2005) use mathematical models like Vector Space Model (VSM) (Salton et al., 1975), Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999), and LDA, in order to capture the topical aspects of the essay.

3 Data Sets

We focus on two tasks: distinguishing full papers (a) from short papers; and (b) from (long) workshop papers, at ACL conferences.

For task (a), we downloaded the full papers and short papers published at the ACL conference from the year 2008 to 2013 from the ACL anthology[2]. Those specific ACL conferences are conferences that have short papers (as indicated by the ACL anthology). Papers in 2008-2011 are used as training data, while papers in 2012 and 2013 are testing data.

For task (b), using the same data set as in (Bergsma et al., 2012), we downloaded the full papers and workshop papers published at the ACL conference and in its joint workshops from the year 2001 to 2009. Papers in 2001-2007 are training data, the rest are testing data. We filtered out workshop papers shorter than 8 pages, since full papers usually comprise from 8 to 10 pages. [3]

Whereas we normalize for length of papers for both tasks. The reason of this normalization is that we want uncover the differences in the content of the papers themselves, since when an author prepares a paper she/he doesn't necessarily know a priori whether the work is more appropriate for short or full paper. Additionally, in other areas of CS, lengths of different types of papers (e.g. full vs breaking) are the same, hence factoring out length will help in porting our work to those other areas. The statistics of the data sets are

[1] http://clair.eecs.umich.edu/aan/

[2] http://aclweb.org/anthology/

[3] In (Bergsma et al., 2012), they filtered out documents with fewer than 100 sentences, similarly to what we are doing here.

shown in Table 1,2. We did not utilize the parsed papers from the ACL Anthology Reference Corpus (ARC)[4] or from the AAN. The ARC only contains papers up to the year 2007. We found that the AAN metadata is sometimes inaccurate, as concerns both authors' and authors' affiliations. The parsed XML text does not contain all the structure information we need (e.g. number of pages).

Years	Full Papers	Short Papers
2008-2011	564	358
2012-2013	285	229

Table 1: Count of Full/Short Papers

Years	Full Papers	Workshop Papers
2001-2007	576	716
2008-2009	240	196

Table 2: Count of Full/Workshop papers

4 Approach

Our system comprises three main components: (1) paper parser; (2) feature extractors; and (3) modeling and evaluation component. The architecture is shown in Figure 1.

Papers are first parsed into structured data (Java objects). Metadata and terms from the paper are used directly as features (see the top two boxes on the right hand-side of Figure 1). The parsed papers are further processed by the four feature extractors in the bottom right of Figure 1. Finally, the feature vectors are sent to supervised learning algorithms.

Here we briefly summarize the six types of features we employ, which we will describe in detail in the rest of this section.

- **Metadata (#TAB, #FMLA, #FIG)** consists of the information extracted from the paper itself. We use the normalized number of tables/formulas/figures per page as features.
- **Terms TF-IDF Score (TITLE, ABSTR)**: The maximum TF-IDF scores of terms in the title and abstract.
- **Sentence complexity (COMPX)**: A vector of the distribution of sentences by their syntactic complexity, as indicated by the depth of their phrase structure tree generated by the Stanford Parser (Klein and Manning, 2002).
- **LDA (Latent Dirichlet Allocation) (TOPIC)**: LDA is used to extract topics.
- **Popular techniques terms (TECHT)**: The counts of techniques which frequently appear in all years of ACL conferences we consider.
- **Author ranking (RANK)**: A list of authors extracted from Microsoft Academic Search[5].

4.1 Paper Parser

The paper parser consists of three subcomponents: Apache Tika[6], Metadata Extractor, and Title & Author Extractor.

The original papers downloaded from the ACL web-site are first sent to Apache Tika, which parses PDF files into HTML-like structured data. The paper is parsed into several <page>tags which contain <p>tags that denote the raw paragraphs in the original paper.

Because the parsing is not very accurate, the raw paragraphs contain all the text in the paper, including the page footer, like "Proceedings of...", "BioNLP 20...". After all these noise sentences are discarded by regular expression matching, the first raw paragraph is considered as the candidate title. The paragraph that starts with "abstract" is taken to be the abstract paragraph. If no paragraph starts with the term "abstract", the first paragraph that contains more than 300 English letters is considered as the abstract paragraph. We consider all the paragraphs between title and abstract as raw author information that contains the author names, affiliations, and email addresses. All paragraphs after the key-words "reference" or "references" are considered as paper references. All the paragraphs between abstract and references are considered as content paragraphs. Content paragraphs are then sent to the Metadata Extractor. The Metadata Extractor uses regular expressions `"^(fig\.|figure)\s*\d+"` and `"^(tab\.|table)\s*\d+"` to find the number of figures **(#FIG)** and number of tables **(#TAB)** on each page. Continuous paragraphs which contain more than 30% non-alphabet English letters are considered as one formula. **#FMLA** is computed as the number of lines which have been identified as formulas rather than number of formulas itself.

[4] http://acl-arc.comp.nus.edu.sg/

[5] http://academic.research.microsoft.com/RankList?entitytype=2&topDomainID=2&subDomainID=9

[6] Apache Tika, http://tika.apache.org/

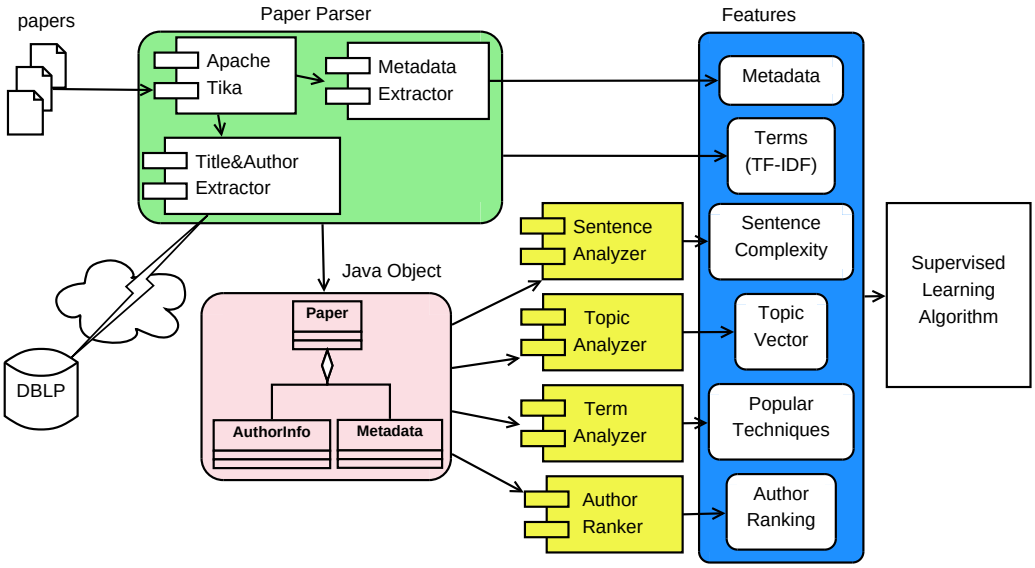

Figure 1: The architecture of the proposed approach

Candidate title and raw author information are processed by the Title & Author Extractor. Because we want our system to have high parsing accuracy, we utilize the DBLP's web service API[7]. This API supports fuzzy query search. We send the raw title to the API and DBLP returns a list of matches (if any) ordered by relevance. We take the real title for the paper to be the highest ranked match returned by DBLP. Together with the title, author names are also returned by the DBLP web API. If a paper does not exist in the DBLP[8], we use Tika to extract the "TITLE" and "AUTHOR" metadata information from the paper. Such function of Tika is based on a standard part of metadata models, numbered ISO-11179[9]. If again no title and/or author names are found in the paper's metadata, we accept the raw title as the title of the paper, and the line following the raw title as author names. The text between the author name and email addresses are considered as the author's affiliation (email addresses are recognized via regular expressions).

The final product of the paper parser, is the PDF paper parsed into a structured Java "Paper" object that contains one "Metadata", several "Author" objects, all the sentences in the abstract/content paragraphs, and all the references.

It is plausible to assume that for a paper to appear at ACL, it must provide a novel contribution, even if lesser novelty is expected of short papers. At the same time, long and short papers must be relevant, i.e., topic and the techniques they use must fall within the purview of the conference itself. Relevance is often judged even more stringently for workshops, since they often address very specific topics; at the same time, workshop papers are not usually as stringently judged on novelty, even if they potentially may address novel areas. To strike a balance between these seemingly contrasting constraints, we capture novelty with respect to title and abstract, under the assumption that papers with novel ideas may have titles/abstracts that contain infrequent terms. In contrast, we capture relevance with respect to the body of the paper (via the topic and technique analysis to be discussed in Section 4.2). After the paper has been parsed, TF-IDF scores for title/abstract terms are computed. Specifically, stemmed non-stop terms in the title only are used to compute the TF-IDF score for the title, and likewise for the abstract. The highest score of terms in the title and abstract is used as the features **TITLE** and **ABSTR** respectively.

The reader may be wondering why we do not use citations as part of our features. Preliminary experiments showed that including citations did not help. We will return to this issue in the Conclusions.

[7] http://www.dblp.org/search/api/?q=[TITLE_TO_SEARCH]

[8] If our model were to be used to assess whether a potential submission reaches the quality of a full paper, of course the paper would not appear **yet** in DBLP – DBLP can be easily bypassed in our architecture.

[9] ISO-11179, http://metadata-stds.org/11179/

4.2 Feature Extractors

Because there is a limit on the number of pages allotted to conference papers, accepted papers must be clear, expressive and reach a certain level of writing proficiency. Based on this assumption, we use a **sentence analyzer** to analyze the complexity of sentences in each paper. Paragraphs in the paper are split into sentences first, which are then parsed by the Stanford Parser. The depth of the sentence parse tree is used as the measure of sentence complexity. Traditional measures of sentence complexity (e.g. to assess the reading level of a text) simply use word and/or sentence length (Swales, 1990; Posteguillo, 1999). More sophisticated measures, including parse tree height, have recently been used in computational models, e.g. (Schwarm and Ostendorf, 2005). As mentioned, the most useful syntactic features in (Bergsma et al., 2012) are reranking features; however, they note those are very computationally demanding. We chose a middle path as far as encoding syntactic complexity, and the experiment result shows our model works better than theirs on the same dataset. For each paper, we count the number of sentences with complexity 1, 2, ... up to 40. We build a feature vector **(COMPX)** of length 40, where item i represents the frequency of sentences with complexity i, namely, the number of sentences with complexity i divided by the total number of sentences in that paper.

Each discipline is characterized by a certain number of core topics. Within that purview, more specific topics come into focus. We believe that even the research interests and techniques are shifting, the research topics are remain relatively stable, and indirectly characterize relevance for that discipline, and that conference. In order to find out which topics have more distinguishing power for a certain year, the **topic analyzer** uses LDA (Latent Dirichlet Allocation). We extract the top N topics from each training data set; the probability distribution vector of these N topics is used as the feature **TOPIC**. We experimented with various values for N (N=5, 10, 15, ..., 75), and the best N is selected by cross validation for each data set. We also experimented with Hierarchical Dirichlet Processes (HDP) (Teh et al., 2006) to eliminate the need to find the best N. However, the results with HDP were worse than when selecting the best N as just described. Hence, in this paper we will not discuss HDP further.

Unlike the topic analyzer, the **technical term analyzer** tries to capture technical concepts related to the methodology employed by the paper: algorithms, techniques, metrics. The LDA topics just discussed are latent, should closely characterize each data set, and their number is one of the experimental parameters. To the contrary, the list of technical concepts is a set list of relevant phrases. For this reason, we build it across all years, and using all papers. The contents of all the papers are parsed and chunked into phrases using OpenNLP [10]. All the noun phrases (NPs) are gathered. Because technical terms mentioned in academic papers most often consist of nouns/abbreviations which contain upper-case letters, as opposed to all lower-case letters, we discard NPs that contain numbers or symbols, possessive symbols ('s, s'), or the conjunction "and". Among the remaining NPs, all phrases containing upper-case characters but not in the first term (e.g. "limited-memory BFGS") are considered as a candidate technique term (we exclude phrases whose first term is capitalized because the capitalization is most likely due to being the first word in a sentence). Acronyms (single terms with all letters in upper-case, e.g. "SVM") are considered as candidate technique terms as well.

These filters result in 57128 candidates, which are ordered in decreasing order of number of papers that contain them. The distribution follows Zipf's law. Terms that appear in fewer than 10 papers are filtered out. This leaves 285 candidates. A manual step takes place at this point: we check the list of candidates to select those that truly correspond to algorithms / models / methodology / metrics. In the end, we are left with 41 technique terms that cover 97 potential candidates, since 56 of those techniques are referred to with acronyms and/or with synonyms (for example, MaxEnt and Maximum Entropy). The most frequent metric term is "TER", that appears in 1298 papers, while the most frequent algorithm term is "SVM" which appears in 660 papers. while the least frequent term (LCS, longest common subsequence) appears in 38 papers. The feature vector **TECHT** for a paper contains the number of occurrences of each of the 41 technique terms.

The **author ranker** extracts the top 2000 authors in the "Natural Language & Speech" area from Microsoft Academic Search. Then the au-

[10] OpenNLP, http://opennlp.apache.org/

thor names, affiliations and ranking are stored in a Lucene index[11]. Because abbreviation of names may appear in the paper, we send a fuzzy query to the Lucene index to retrieve the rank information. The query is constructed following these rules: (1) for the name, at least one of first name, last name, and middle name (if it exists) must be matched to the record in the index; (2) at least one of the non-stop words in the affiliation must be matched. (We tried to assign higher weight to last name than to first and middle names during the matching, however the overall model accuracy decreases.) The records retrieved from the index against the query are ordered in descending order of matching scores. An empirical matching score threshold is set in order to filter out unlikely matches. Since we do not know how many authors of papers exist in the list of top 2000 ranked researchers, the threshold is tuned mainly based on the precision of matching which finally reaches 96.03%, for a total of 474 unique authors matched in that list.

The rank of the best matched record is retrieved as the author's rank. The best rank among all the authors of the paper is used as the feature, **RANK**.

5 Results and Discussion

We experimented with four supervised approaches: Decision Trees (C4.5), Naive Bayes, Logistic Regression and SVM. SVM models performed better in every single experiment, statistically significantly so when compared to baseline and to each of the other algorithms (via χ^2 at the 0.05% significance level). Hence, in the following we only report SVM results.

As mentioned in Section 3 we have two training data sets, and two testing data sets for two different tasks. We built separate models that distinguish between full and short papers, and full and workshop papers independently on those two training data sets. Since there are 9 proposed features in our system, the number of possible combinations is $2^9 - 1 = 511$. We evaluate our approach and select the best feature combination by conducting 10-fold cross-validation using all 511 feature combinations on each of the two training data sets. Additionally, we experiment with different numbers of LDA topics, with N varying over $\{5, 10, 15, ..., 75\}$, and find that on both training data sets, the best N is 35. Then, we test our models on the corresponding testing data sets.

[11] Apache Lucene, http://lucene.apache.org/

For distinguishing full papers from short papers, since no previous results exist as far as we know, the baseline is obtained by randomly assigning full or short to each paper, following the original distribution. This sample procedure was iterated for 99 rounds. The result with the median F-score among the 99 iterations is used as our baseline ($F1 = 55.7\%$). On testing data set, we compare our result with the best Venue task performance $F1 = 66.7\%$ in (Bergsma et al., 2012),

Table 3 shows the results on testing data using best features which are selected from training data sets. We can see from the results that our model work better than all the baselines, and outperform (Bergsma et al., 2012)'s model by at least 3.6%. For the 'vs workshop' task, besides the best feature combination, we also list the best combination without RANK feature. This is because ACL conference submissions are double blind, in practice, we may not get such ranking information. Interestingly, this second feature combination does not include TECHT features either. TECHT features are the only ones in our model where some manual filtering was applied. Hence, we show that we can obtain better models with a completely automatic pipeline as well.

Turning now to the features that appear in the best feature combinations, metadata as concerns the number of lines containing formulas (#FMLA) appears in all the three models. Number of tables per page (#TAB) and sentence complexity (COMPX) are useful in differentiating full papers from short papers and from workshop papers when RANK is not available. Measures of "completeness/evaluation" of the research as expressed by number of tables, figures, and/or formulas are important to distinguish full from (long) workshop papers at ACL. As far as COMPX is concerned, we will show later that there may be latent models that full paper fit better than the others (but contrary to our initial intuition, it is short/workshop papers that have more complex sentences). TOPIC appears in every combination for the 'vs workshop' task, but does not work so well for the 'vs short' task. This corresponds to the intuition that full papers resemble short papers more than workshop papers as far as topic is concerned (which is more focused for workshops).

We also compared the performance of each individual feature (due to space limitation, we don't list them here). The result shows that even if

Task	Pre.	Rec.	F-1	Base.	Best Feature Combination
vs Short	75.1	74.5	**74.6**	55.7	#FMLA+#TAB+ABSTR+COMPX
vs Workshop	72.1	72.0	**71.7**	66.7*	#FMLA+TITLE+ABSTR+TOPIC+TECHT+RANK
	71.1	70.9	**70.3**		#FMLA+#TAB+TITLE+TOPIC+COMPX

Table 3: Best Feature Combination on Testing Data. ** is the best result reported in (Bergsma et al., 2012)*

TITLE, ABSTR and RANK don't perform well when used alone ($F1 = 55.6\%, 56.5\%, 41.5\%$ on 'vs workshop' testing data set, respectively), as we can see, they appear in the best feature combinations. Recall that the TITLE/ABSTR feature attempts at capturing novelty via the highest TF-IDF score for terms in the title/abstract. Since reviewing is double blind for full and short papers, we hypothesize that the RANK feature appearing in some results simply reflects the quality of work that highly ranked authors produce. It is also possible that reviewers guess some authors' identity, i.e. via the topic or references in the paper. However this possibility seems remote, since RANK by itself is not very predictive.

5.1 Model analysis

The following features appear in more than one best model: Metadata (#FMLA, #TAB), TOPIC, TF-IDF scores (ABSTR, TITLE) and sentence complexity (COMPX). Hence, they deserve further investigation. Additionally, we investigate the role played by RANK, since we want to assess why it appears in the "full vs workshop" models, but not in the "full vs short" models. For LDA topics, plausible interpretations of 4 out of the 5 most distinguishing topics in "vs workshop" model (topic is not among best features for "vs short" model) are: dialogue/speech; ontology and semantics; annotation; probabilistic models. We leave the further study of the effect of LDA for future work, since they are latent (e.g. probabilistic distributions over words).

Author ranking (RANK) We did not find differences between the respective ranks of authors in the full, short and workshop papers: namely, each year, the highest, lowest, and average ranks of authors of full, short and workshop papers are very similar. However, the number of papers for which at least one author appears in the top author list are different in the 'vs workshop' group, but not in the 'vs short' group. Table 4 shows that among all the papers in our data sets, around half of the full/short papers have top authors, while only one fourth of workshop papers do. This seems to suggest that highly ranked authors value maintaining their research paths more than moving to new topics, which are more often addressed by workshops. It also explains why RANK does not appear among predictive features for the 'vs short' task.

	Full	Short	Workshop
Highest rank	2	2	2
Lowest rank	1997	1997	1978
Average rank	580	586	583
Percentage (%)	53.0	44.3	25.1

Table 4: Highest/lowest/average author ranks, and percentage of papers where at least one author belongs to the top 2000 author list

Feature	Full	Short	Workshop
#FMLA	2.356	1.639	1.327
#TAB	0.426	0.455	0.378

Table 5: Avg. number of formulas, and tables per page

Metadata (#FMLA, #TAB) Table 5 shows the statistics for the average number of each metadata type on each page in each data set – tables, formulas (recall that #FMLA is the number of lines that have been identified as formulas). It can be seen that full papers always include more formula lines than short/workshop papers, and almost so for tables. We speculate that tables and formulas correlate with mature work, since e.g. they characterize more extensive evaluation. Additionally, tables may be indicative of better written papers since they help summarize concepts that may be difficult to effectively express (only) in words.

TF-IDF scores (TITLE, ABSTR) As mentioned, we use the maximum TF-IDF scores of terms in the title and abstract as features in our system. We measured the average value of these two features for different paper types, see Table 6. Full papers seem to use more novel terms in the abstract than others, but on the other hand use fewer novel terms in title. This may suggest that full papers focus more on mainstream research topics (as reflected by the title), but use novel tech-

niques (as reflected by the abstract). On the contrary, short/workshop papers address more novel topics but use more mature techniques.

Feature	Full	Short	Workshop
TITLE	6.95	7.30	7.63
ABSTR	17.15	15.55	15.82

Table 6: Avg. TITLE/ABSTR in different paper types

Data Set	#SigCompl	#SigFull
vs short	35	33
vs workshop	35	31

Table 7: Fit to model of sentence complexity

Sentence complexity (COMPX) We believe that the difference in sentence complexity between full and short papers, and also wrt workshop papers to a smaller extent, may be due to a latent model. As we mentioned in Section 4.2, for each paper j, $x_{i,j}$ is the frequency of sentences.

To verify whether a latent model underlying sentence complexity exists, first, for each complexity i we build two vectors, the distribution of the positive sample ($full_{i,1}, full_{i,2}, ..., full_{i,FULL}$) and of the negative sample ($other_{i,1}, other_{i,2}, ..., other_{i,OTHER}$) (where FULL and OTHER are the total number of full and short/workshop papers, respectively). Hence, there are 40 pairs of such vectors. For each complexity i, we check whether the two distributions are significantly different via the F-test (α=0.01). If they are, we compute the variances for both positive (full) and negative (short/workshop) samples, to see if variance among full papers at complexity i is smaller than the variance among short/workshop papers (i.e. it fits the latent model more tightly). The results are shown in Table 7. *#SigCompl* is the number of complexity values that are significantly different, out of 40 possible, in our data sets; *#SigFull* is the number of *#SigCompl* complexity values for which the full papers fit the model more tightly. E.g. for 'vs short', on 35 out of 40 complexity values, the two distributions are different (*#SigCompl = 35*); 33 times out of those 35 (*#SigFull=33*), the variance among full papers is lower than the variance for short papers. Contrary to our initial intuition, it is full papers that on average have the lowest level of complexity, workshop papers in the middle, and short the highest. Since short papers have less space, authors seem to pack as much information as possible, including via syntactic structures.

6 Conclusions and Future Work

Our results show that we can assess a paper's quality with good results, we outperform (Bergsma et al., 2012)'s model by at least 3.6%. Topic distribution as learned by the LDA model, metadata (number of tables, and number of lines of mathematical formulas) and sentence complexity are the most predictive features. TF-IDF of title and abstract, author ranking, and technique terms also contribute to the overall models. Topics and technique terms indirectly capture the notion of relevance and presumably also specific trends in research. On the other hand, TF-IDF of title and abstract partly reflect novelty. Metadata hints at the fact that mature work is described in a more structured way. Full papers also fit the model of sentence complexity more tightly. Because reviewing is blind, the author's ranking cannot directly influence the decision of the reviewers. However, author ranking probably reflects the quality of work that highly ranked authors produce.

There are many open venues for improvement. First, we intend to apply our methodology to other conferences in the same area, and in other areas of Computer Science. We intend to enhance the technique term extraction procedure, which will eventually eliminate the only manual step in the whole pipeline. The novelty of a paper usually plays a very important role in the perceived quality of a paper, so we will investigate more sophisticated ways to measure such novelty. In this work, we did not use the paper references among our features, since preliminary experiments that included references among the features were not promising. However, because a good work tends to cite good works, we will explore more sophisticated models of citations, such as those based on citation networks (Mohammad et al., 2009).

References

McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu. 2002.

Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric Analysis of Scientific Articles. In *Proceedings of the Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, pages 327–337, Montréal, Canada, June. Association for Computational Linguistics.

Steven Bethard and Dan Jurafsky. 2010. Who should I cite: learning literature search models from citation behavior. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 609–618. ACM.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Jill Burstein, Martin Chodorow, and Claudia Leacock. 2003. CriterionSM: Online essay evaluation: An application for automated evaluation of student essays. In *Proceedings of the Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence*.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics.

Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 50–57. ACM.

Tuomo Kakkonen, Niko Myller, Erkki Sutinen, and Jari Timonen. 2005. Comparison of dimension reduction methods for automated essay grading. *Natural Language Engineering*, 1:1–16.

Dan Klein and Christopher D. Manning. 2002. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15*, pages 3–10.

Lisa McGrath and Maria Kuteeva. 2011. Stance and engagement in pure mathematics research articles: Linking discourse features to disciplinary practices. *English for Specific Purposes*.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using Citations to Generate Surveys of Scientific Paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 584–592, Boulder, Colorado, June. Association for Computational Linguistics.

Ellis B. Page. 1966. The Imminence of... Grading Essays by Computer. *Phi Delta Kappan*, pages 238–243.

Santiago Posteguillo. 1999. The schematic structure of Computer Science research articles. *English for Specific Purposes*, 18(2):139–160.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Sarah E. Schwarm and Mari Ostendorf. 2005. 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 435–442.

John Swales. 1990. *Genre Analysis: English in Academic and Research Settings*. Cambridge University Press.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2006. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476).

Dani Yogatama, Michael Heilman, Brendan O'Connor, Chris Dyer, Bryan R. Routledge, and Noah A. Smith. 2011. Predicting a scientific community's response to an article. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 594–604. Association for Computational Linguistics.