Extracting argument and domain words for identifying argument components in texts

Huy V. Nguyen
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15260, USA
hvn3@pitt.edu

Diane J. Litman
Computer Science Department & LRDC
University of Pittsburgh
Pittsburgh, PA 15260, USA
litman@cs.pitt.edu

Abstract

Argument mining studies in natural language text often use lexical (e.g. n-grams) and syntactic (e.g. grammatical production rules) features with all possible values. In prior work on a corpus of academic essays, we demonstrated that such large and sparse feature spaces can cause difficulty for feature selection and proposed a method to design a more compact feature space. The proposed feature design is based on post-processing a topic model to extract argument and domain words. In this paper we investigate the generality of this approach, by applying our methodology to a new corpus of persuasive essays. Our experiments show that replacing n-grams and syntactic rules with features and constraints using extracted argument and domain words significantly improves argument mining performance for persuasive essays.

1 Introduction

Argument mining in text involves automatically identifying argument components as well as argumentative relations between components. Argument mining has been studied in a variety of contexts including essay assessment and feedback (Burstein et al., 2003; Stab and Gurevych, 2014b), visualization and search in legal text (Moens et al., 2007), and opinion mining in online reviews and debates (Park and Cardie, 2014; Boltužić and Šnajder, 2014). Problem formulations of argument mining have ranged from argument detection (e.g. does a sentence contain argumentative content?) to argument component (e.g. claims vs. premise) and/or relation (e.g. support vs. attack) classification.

Due to the loosely-organized nature of many types of texts, associated argument mining studies have typically used generic linguistic features, e.g. n-grams and syntactic rules, and counted on feature selection to reduce large and sparse feature spaces. For example, in texts such as student essays and product reviews there are optional titles but typically no section headings, and claims are substantiated by personal experience rather than cited sources. Thus, specialized features as used in scientific articles (Teufel and Moens, 2002) are not available.

While this use of generic linguistic features has been effective, we propose a feature reduction method based on the semi-supervised derivation of lexical signals of argumentative and domain content. Our approach was initially developed to identify argument elements, i.e. hypothesis and findings, in academic essays (written following APA guidelines) of college students (Nguyen and Litman, submitted). In particular, we post-processed a topic model to extract argument words (lexical signals of argumentative content) and domain words (terminologies in argument topics) using seeds from the assignment description and essay prompts. The extracted argument and domain words were then used to create novel features and constraints for argument mining, and significantly outperformed features derived from n-grams and syntactic rules.

In this paper we apply our argument and domain word extraction method to a new corpus of persuasive essays, with the goal of answering: (1) whether our proposed feature design is general and can be

Proceedings of the 2nd Workshop on Argumentation Mining, pages 22–28,
Denver, Colorado, June 4, 2015. ©2015 Association for Computational Linguistics

(1) My view is that the [*government should give priorities to invest more money on the basic social welfares such as education and housing instead of subsidizing arts relative programs*]$_{majorClaim}$. ¶
(2) [*Art is not the key determination of quality of life, but education is*]$_{claim}$. (3) [*In order to make people better off, it is more urgent for governments to commit money to some fundamental help such as setting more scholarships in education section for all citizens*]$_{premise}$ ... ¶
(4) To conclude, [*art could play an active role in improving the quality of people's lives*]$_{premise}$, but I think that [*governments should attach heavier weight to other social issues such as education and housing needs*]$_{claim}$ because [*those are the most essential ways enable to make people a decent life*]$_{premise}$.

Figure 1: Excerpt of a persuasive essay with three paragraphs. The title is "*Do arts and music improve the quality of life?*". Sentences are numbered for easy look-up. Argument components are enclosed in square brackets.

adapted easily across different corpora, (2) whether lexical signals of argumentative content (part of our proposed features) learned from one corpus also signal argumentation in a second corpus. For the first question we test whether features based on argument and domain words outperform n-grams and syntactic rules for argument mining in persuasive essays. For the second question, we test whether our originally derived argument word set is useful for argument mining in persuasive essays.

2 Data

Data for our study is an annotated corpus of persuasive essays[1] (Stab and Gurevych, 2014a). Writing prompts of persuasive essays requires students to state their opinions (i.e. major claims) on topics and validate those opinions with convincing arguments (i.e. claim and premise). Figure 1 shows an excerpt of an annotated persuasive essay in the corpus.

The corpus consists of 1673 sentences in 90 essays collected from www.essayforum.com. Essay sentences were annotated for possible argument components of three types: *major claim* – writer's stance towards the topic, *claim* – controversial statement that supports or attacks major claim, and *premise* – underpins the validity of claim. An argument component can be a clause, e.g. premises in sentence (4), or the whole sentence, e.g. claim sentence (2). A sentence can have from zero to multiple argument components (yielding more data instances than corpus sentences). Inter-rater agreement of three annotators was α_U = 0.72.

MajorClaim	Claim	Premise	None
90	429	1033	327

Table 1: Number of instances in each class.

Class distribution of total 1879 instances is shown in Table 1. Except for the *None* class which consists of 327 sentences having no argument component, the other classes contain the exact argument components so their instances can be clauses or sentences (Stab and Gurevych, 2014b).

3 Prediction Models

3.1 Baseline

Stab and Gurevych (2014b) utilized the corpus (Stab and Gurevych, 2014a) for automated argument component identification. We re-implement their features as a baseline to evaluate our approach.

Structural features: #tokens and #punctuations in argument component (AC), in covering sentence, and preceding/following the AC in sentence, token ratio between covering sentence and AC. Two binary features indicate if the token ratio is 1 and if the sentence ends with a question mark. Five position features are sentence's position in essay, whether the AC is in the first/last paragraph, the first/last sentence of a paragraph.

Lexical features: all n-grams of length 1-3 extracted from AC's including preceding text which is not covered by other AC's in sentence, verbs like '*believe*', adverbs like '*also*', and whether the AC has a modal verb.

Syntactic features: #sub-clauses and depth of parse tree of the covering sentence, tense of main verb and production rules (VP → VBG NP) from parse tree of the AC.

Discourse markers: discourse connectives of 3 relations: comparison, contingency, and expansion but not temporal[2] extracted by addDiscourse program (Pitler et al., 2009).

[1] A type of writing response to test questions on standardized tests (cf. (Burstein et al., 2003)).

[2] Authors of (Stab and Gurevych, 2014b) manually collected 55 PDTB markers after removing those that do not indicate argumentative discourse, e.g. markers of temporal relations.

First person pronouns: whether each of *I*, *me*, *my*, *mine*, and *myself* is present in the sentence.

Contextual features: #tokens, #punctuations, #sub-clauses, and presence of modal verb in preceding and following sentences.

3.2 Proposed model

Our proposed model is based on the idea of separating argument and domain words (Nguyen and Litman, submitted) to better model argumentative content and argument topics in text. It is common in argumentative text that argument expressions start with an argument shell[3], e.g. "*My view is that*", "*I think*", "*to conclude*" followed by argument content. To model this writing style, we consider features of lexical and structural aspects of the text. As for the *lexical aspect*, we learn a topic model using development data (described below) to separate argument words (e.g. '*view*', '*conclude*', '*think*') from domain words (e.g. '*art*', '*life*'). Compared to n-grams, our argument words provide a much more compact representation. As for the *structural aspect*, instead of production rules, e.g. "*S → NP VP*", we use dependency parses to extract pairs of subject and main verb of sentences, e.g. "*I.think*", "*view.be*". Dependency relations are minimal syntactic structures compared to production rules. To further make the features topic-independent, we keep only dependency pairs that do not include domain words.

3.2.1 Post-processing a topic model to extract argument and domain words

We define argument words as those playing a role of argument indicators and commonly used in different argument topics, e.g. '*reason*', '*opinion*', '*think*'. In contrast, domain words are specific terminologies commonly used within the topic, e.g. '*art*', '*education*'. Our notions of argument and domain languages share a similarity with the idea of shell language and content in (Madnani et al., 2012) in that we aim to model the lexical signals of argumentative content. However while Madnani et al. (2012) emphasized the boundaries between argument shell and content, we do not require such a physical separation between the two aspects of an argument. Instead we emphasize more the lexical signals themselves and allow argument words to occur in the argument content. For example, the major claim in Figure 1 has two argument words '*should*' and '*instead*' which makes the statement controversial.

To learn argument and domain words, we run the LDA (Blei et al., 2003) algorithm[4] and post-process the output. Our development data to build the topic model are 6794 essays posted on www.essayforum.com excluding those in the corpus. Our post-processing algorithm requires a minimal seeding with predefined argument keywords and essay prompts (i.e. post titles). We examine frequent words (more than 100 occurrences) in prompts of development data and choose 10 words as argument keywords: *agree, disagree, reason, support, advantage, disadvantage, think, conclusion, result* and *opinion*. Seeds of domain words are those in the prompts but not argument or stop words. Each domain seed word is associated with an occurrence frequency f as a ratio of the seed occurrences over total occurrences of all domain seeds in essay prompts. All words including seeds are then stemmed.

We vary the number of LDA topics from 20 to 80; in each run, we return the top 500 words for each topic, then remove words with total occurrence less than 3. For words in multiple LDA topics, we compare every pair of word probability given each of two topics t_1, t_2: $p(w|t_1)$ and $p(w|t_2)$ and remove the word from topic with smaller probability if the ratio $p(w|t_1)/p(w|t_2) > 7$. This allows us to only punish words with very low conditional probability while still keeping a fair amount of multiple-topic words.

For each LDA topic we calculate three weights: argument weight (AW) is the number of unique argument seeds in the topic; domain weight (DW) is the sum of frequencies f of domain seeds in the topic; and combined weight $CW = AW - DW$. To discriminate the LDA topic of argument words from LDA topics of domain words given a number of LDA topics, we compute a relative ratio of the largest over the second largest combined weights (e.g. $(CW_{t1} - CW_{t2})/CW_{t2}$ as in Table 2). These settings prioritize argument seeds and topics with more argument seeds, and less domain seeds. Given the number of LDA topics that has the highest ratio (36 topics given our development data), we select LDA topic with the largest combined weight as the

[3] Cf. shell language (Madnani et al., 2012)

[4] We use GibbsLDA++ (Phan and Nguyen, 2007)

Topic 1 *reason exampl support agre think becaus disagre statement opinion believe therefor idea conclus*
Topic 2 *citi live big hous place area small apart town build communiti factori urban*
Topic 3 *children parent school educ teach kid adult grow childhood behavior taught*

Table 2: Samples of top argument (topic 1), and domain (topics 2 and 3) words. Words are stemmed.

argument word list. Domain words are the top words of other topics, but not argument or stop words.

Table 2 shows examples of top argument and domain words (stemmed) returned by our algorithm. Given 10 argument keywords, our algorithm returns a list of 263 argument words which is a mixture of keyword variants (e.g. *think, believe, viewpoint, opinion, argument, claim*), connectives (e.g. *therefore, however, despite*), and other stop words.

Our proposed model takes all features from the baseline except n-grams and production rules, and adds the following features: *argument words* as unigrams, *filtered dependency pairs* (§3.2) as skipped bigrams, and *numbers* of argument and domain words.[5] Our proposed model is compact with 956 original features compared to 5132 of the baseline[6].

4 Experimental Results

4.1 Proposed vs. baseline models

Our first experiment replicates what was conducted in (Stab and Gurevych, 2014b). We perform 10-fold cross validation; in each run we train models using LibLINEAR (Fan et al., 2008) algorithm with top 100 features returned by the InfoGain feature selection algorithm performed in the training folds. We use LightSIDE (lightsidelabs.com) to extract n-grams and production rules, the Stanford parser (Klein and Manning, 2003) to parse the texts, and Weka (Hall et al., 2009) to conduct the machine learning experiments. Table 3 (left) shows the performances of three models.

	BaseR	BaseI	AD	BaseI	AD
#features	100	100	100	130	70
Accuracy	0.77	0.78	0.79+	0.80	0.83*
Kappa	NA	0.63	0.65*	0.64	0.69*
F1	0.73	0.71	0.72	0.71	0.76+
Precision	0.77	0.76	0.76	0.76	0.79
Recall	0.68	0.69	0.70	0.68	0.74+
F1:MajorClaim	0.62	0.54	0.51	0.48	0.59
F1:Claim	0.54	0.47	0.53*	0.49	0.56*
F1:Premise	0.83	0.84	0.84	0.86	0.88*
F1:None	0.88	1.00	1.00	1.00	1.00

Table 3: Model performances with top 100 features (left) and best number of features (right). +, * indicate $p < 0.1, p < 0.05$ respectively in AD vs. BaseI comparison.

We note that there are notable performance disparities between BaseI (our implementation §3.1), and BaseR (reported performance of the model by Stab and Gurevych (2014b)). Particularly, BaseI obtains higher F1:Premise, F1:None, and smaller F1:MajorClaim, F1:Claim than BaseR. The differences may mostly be due to dissimilar feature extraction methods and NLP/ML toolkits. Comparing BaseI and AD (our proposed model using learned **a**rgument and **d**omain words §3.2, §3.2.1) shows that our proposed model AD yields higher Kappa, F1:Claim (significantly) and accuracy (trending).

To further analyze performance improvement by the AD model, we use 75 randomly-selected essays to train and estimate the best numbers of features of BaseI and AD (w.r.t F1 score) through a 9-fold cross validation, then test on 15 remaining essays. As shown in Table 3 (right), AD's test performance is consistently better with far smaller number of top features (70) than BaseI (130). AD has 6 of 31 argument words not present in BaseI's 34 unigrams: *analyze, controversial, could, debate, discuss, ordinal*. AD keeps only 5 dependency pairs: *I.agree, I.believe, I.conclude, I.think* and *people.believe* while BaseI keeps up to 31 bigrams and 13 trigrams in the top features. These indicate the dominance of our proposed features over generic n-grams and syntactic rules.

4.2 Alternative argument word list

In this experiment, we evaluate the prediction transfer of the actual argument word list across genres. In (Nguyen and Litman, submitted), our LDA post-processing algorithm returned 429 argument words from a development set of 254 academic writings, where the seeds (*hypothesis, support, opposition, finding, study*) were taken from the assignment. To

[5] A model based on seed words without expansion to argument words yields significantly worse performance than the baseline. This shows the necessity of our proposed topic model.

[6] N-grams and production rules of less than 3 occurrences were removed to improve baseline performance.

	AltAD	AD
Accuracy	0.77	0.79*
Kappa	0.62	0.65*
F1:MajorClaim	0.56	0.51
F1:Claim	0.47	0.53*
F1:Premise	0.83	0.84*
F1:None	1.00	1.00

Table 4: Performance with different argument words lists.

build an alternative model (AltAD), we replace the argument words in AD with those 429 argument words, re-filter dependency pairs and update the number of argument words. We follow the same setting in §4.1 to train AD and AltAD using top 100 features. As shown in Table 4, AltAD performs worse than AD, except a higher F1:MajorClaim but not significant. AltAD yields significantly lower accuracy, Kappa, F1:Claim and F1:Premise.

Comparing the two learned argument word lists gives us interesting insights. The lists have 142 common words with 9 discourse connectives (e.g. '*therefore*', '*despite*'), 72 content words (e.g. '*result*', '*support*'), and 61 stop words. 30 of the common argument words appear in top 100 features of AltAD, but only 5 are content words: '*conclusion*', '*topic*', '*analyze*', '*show*', and '*reason*'. This shows that while the two argument word lists have a fair amount of common words, the transferable part is mostly limited to function words, e.g. discourse connectives, stop words. In contrast, 270 of the 285 unique words to AltAD are not selected for top 100 features, and most of those are popular terms in academic writings, e.g. '*research*', '*hypothesis*', '*variable*'. Moreover AD's top 100 features have 20 argument words unique to the model, and 19 of those are content words, e.g. '*believe*', '*agree*', '*discuss*', '*view*'. These non-transferable parts suggest that argument words should be learned from appropriate seeds and development sets for best performance.

5 Related Work

Research in argument mining has explored novel features to model argumentative discourse, e.g pre-defined indicative phrases for argumentation (Mochales and Moens, 2008), headlines and citations (Teufel and Moens, 2002), sentiment clue and speech event (Park and Cardie, 2014). However, the major feature sets were still generic n-grams. We propose to replace generic n-grams with argument words learned using a topic model.

Role-based word separation in texts have been studied in a wide variety of contexts: opinion and topic word separation in opinion mining (see (Liu, 2012) for a survey), domain and review word separation for review visualization (Xiong and Litman, 2013), domain concept word tagging in tutorial dialogue systems (Litman et al., 2009), and dialog act cues for dialog act tagging (Samuel et al., 1998).

Post-processing LDA (Blei et al., 2003) output was studied to identify topics of visual words (Louis and Nenkova, 2013) and representative words of topics (Brody and Elhadad, 2010; Funatsu et al., 2014). Our work is the first of its kind to use topic models to extract argument and domain words from argumentative texts. Our technique has a similarity with (Louis and Nenkova, 2013) in that we use seed words to guide the separation.

6 Conclusions and Future Work

We have shown that our novel method for modeling argumentative content and argument topic in academic writings also applies to argument mining in persuasive essays, with our results outperforming a baseline model from a prior study of this genre.

Our contributions are 2-fold. First, our proposed features are shown to efficiently replace generic n-grams and production rules in argument mining tasks for significantly better performance. The core component of our feature extraction is a novel algorithm that post-processes LDA output to learn argument and domain words with a minimal seeding.

Second, our analysis gives insights into the lexical signals of argumentative content. While argument word lists extracted for different data can have parts in common, there are non-transferable parts which are genre-dependent and necessary for the best performance. Thus such indicators of argumentative content should be learned within genre.

Our next task is argumentative relation classification, i.e. support vs. attack. We would also like to explore sequence labeling to identify argument language, and combine them with topic models.

Acknowledgments. This research is supported by NSF Grant 1122504. We thank Wencan Luo, Zahra Rahimi and the reviewers for their feedback.

References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Filip Boltužić and Jan Šnajder. 2014. Back up your Stance: Recognizing Arguments in Online Discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland, June. Association for Computational Linguistics.

Samuel Brody and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812. Association for Computational Linguistics.

Jill Burstein, Daniel Marcu, and Kevin Knight. 2003. Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems*, 18(1):32–39, January.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *J. Mach. Learn. Res.*, 9:1871–1874, June.

Toshiaki Funatsu, Yoichi Tomiura, Emi Ishita, and Kosuke Furusawa. 2014. Extracting Representative Words of a Topic Determined by Latent Dirichlet Allocation. In *eKNOW 2014, The Sixth International Conference on Information, Process, and Knowledge Management*, pages 112–117.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Diane Litman, Johanna Moore, Myroslava O. Dzikovska, and Elaine Farrow. 2009. Using Natural Language Processing to Analyze Tutorial Dialogue Corpora Across Domains Modalities. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, pages 149–156, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool.

Annie Louis and Ani Nenkova. 2013. What Makes Writing Great? First Experiments on Article Quality Prediction in the Science Journalism Domain. *Transactions of the Association of Computational Linguistics – Volume 1*, pages 341–352.

Nitin Madnani, Michael Heilman, Joel Tetreault, and Martin Chodorow. 2012. Identifying High-Level Organizational Elements in Argumentative Discourse. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 20–28, Montreal, Canada. Association for Computational Linguistics.

Raquel Mochales and Marie-Francine Moens. 2008. Study on the Structure of Argumentation in Case Law. In *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems: JURIX 2008: The Twenty-First Annual Conference*, pages 11–20, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic Detection of Arguments in Legal Texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, ICAIL '07, pages 225–230, New York, NY, USA. ACM.

Huy Nguyen and Diane Litman. submitted. Identifying argument elements in diagram-based academic writing.

Joonsuk Park and Claire Cardie. 2014. Identifying Appropriate Support for Propositions in Online User Comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland, June. Association for Computational Linguistics.

Xuan-Hieu Phan and Cam-Tu Nguyen. 2007. *GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA)*. Technical report.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 683–691. Association for Computational Linguistics.

Ken Samuel, Sandra Carberry, and K. Vijay-Shanker. 1998. Dialogue Act Tagging with Transformation-Based Learning. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 1150–1156, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014a. Annotating Argument Components and Relations in Persuasive Essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014b. Identifying Argumentative Discourse Structures in Persuasive Essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.

Simone Teufel and Marc Moens. 2002. Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics, Volume 28, Number 4, December 2002*.

Wenting Xiong and Diane Litman. 2013. Evaluating Topic-Word Review Analysis for Understanding Student Peer Review Performance. pages 200–207, Memphis, TN, July.