H-0325 (HAI1607-001) July 5, 2016
Computer Science

IBM Research Report

Claim Synthesis via Predicate Recycling

Yonatan Bilu, Noam Slonim
IBM Research Division
Haifa Research Laboratory
Mt. Carmel 31905
Haifa, Israel

Research Division
Almaden – Austin – Beijing – Cambridge – Dublin - Haifa – India – Melbourne - T.J. Watson – Tokyo - Zurich

Claim Synthesis via Predicate Recycling

Yonatan Bilu
IBM Haifa Research Lab
Mount Carmel, Haifa, 31905
Israel
yonatanb@il.ibm.com

Noam Slonim
IBM Haifa Research Lab
Mount Carmel, Haifa, 31905
Israel
noams@il.ibm.com

Abstract

Computational Argumentation has two main goals - the detection and analysis of arguments on the one hand, and the synthesis of arguments on the other. Much attention has been given to the former, but considerably less to the latter.

A key component in synthesizing arguments is the synthesis of claims. One way to do so is by employing argumentation mining to detect claims within an appropriate corpus. In general, this appears to be a hard problem. Thus, it is interesting to explore if - for the sake of synthesis - there may be other ways to generate claims.

Here we explore such a method: we extract the predicate of simple, manually-detected, claims, and attempt to generate novel claims from them. Surprisingly, this simple method yields fairly good results.

1 Introduction

When people argue, how do they come up with the arguments they present, and can a machine emulate this? The motivation for this work comes from this second question, for which the relevant field of study is Computational Argumentation, an emerging field with roots in Computer Science, Mathematics, Philosophy and Rhetorics. However, while much attention is given in the field to the modeling and analysis of arguments, automatic synthesis of arguments receives considerably less.

So, how do people come up with arguments? One way is to read-up on the topic and present the arguments you find in the literature. Another - if the topic at hand is within your field of expertise - is to communicate your opinion. Yet a third way is to "recycle" arguments you are familiar with and apply them to new domains. For example, someone who's concerned about the free speech might use an argument like "it's a violation of free speech" when discussing any one of these topics: whether violent video games should be banned, whether some Internet content should be censored, or whether certain types of advertisement should be restricted.

Argumentation Mining (Mochales Palau and Moens, 2011) is analogous to the first option: Given a corpus, it aims to detect arguments therein (and the relations among them). Thus, it can be used to suggest claims when a relevant corpus is available. The second option is analogous to Natural Language Generation (NLG; (Reiter and Dale, 2000)), where applications such as recommender systems synthesize arguments to explain their recommendations, as done for example in (Carenini and Moore, 2006) .

These approaches yield good results when applied to specific domains. In an NLG application, there is commonly a specific knowledge base which the system communicates. The form and content of arguments are derived and determined by it and are thus limited to the knowledge therein. Similarly, argument mining works well when an argument-rich and topic-related corpus is available - e.g. (Wyner et al., 2010) - but in general seems to be hard (Levy et al., 2014). Thus, it is interesting and challenging to synthesize arguments in an open domain. To the best of our knowledge, this is the first work that directly attempts to address this task.

Modeling of arguments goes back to the ancient Greeks and Aristotle, and more modern work starting perhaps most famously with the Toulmin argument model (Toulmin, 1958). A common element in all such models is the *claim* (or *conclusion*) being forwarded by the argument. Thus, a natural first step in synthesizing arguments in a general setting is being able to synthesize claims in such a setting.

We suggest here a simple way for doing so, based on the aforementioned notion of argument "recycling". Specifically, that the *predicate* of a claim - what it says on the topic at hand - may be applicable to other topics as well. For example, if we are familiar with the claim "banning violent video games *is a violation of free speech*" in the context of the topic "banning violent video games", we could synthesize the claim "Internet censorship *is a violation of free speech*" when presented with the topic "Internet Censorship". The challenge is then to determine whether the synthesized claim is actually coherent and relevant to the new topic, which we do using statistical Machine Learning techniques, as described in Section 2.1.

This two-stages framework - generating text and then selecting whether or not it is appropriate - is reminiscent of Statistical NLG (SNLG; (Langklide and Knight, 1998)). In an SNLG system, after the macro-planning and micro-planning stages (see (Reiter and Dale, 2000)) are executed, and the message to be communicated is determined, multiple candidate realizations are produced, and then statistical methods are used to determine which of these realizations is the best (based on a reference corpus).

Our work differs from SNLG in that there are no pre-determined messages. The generation stage produces candidate *content*. Each candidate claim is a different message, and the selection stage attempts to identify those which are coherent and relevant, rather than best realized. In other words, while the classical NLG paradigm is to first select the content and then realize it in a natural language, here our building blocks from the onset are natural language elements, and statistical methods are used to determine which content selections - implied by combining them - are valid.

Finally, the notion that predicates of claims regarding one topic may be applicable to another is reminiscent of the motivation for the work of (Card et al., 2015), who observe that there are commonalities (so called "framing dimensions") among the way different topics are framed in news articles.

2 Algorithm

The claim synthesis algorithm is composed of three components. The first is a pre-processing component, in which the Predicate Lexicon is constructed. The second is the Generation Component - the input to this component is a topic (and the Predicate Lexicon), and the output is a list of candidate claims. The final component is the Selection Component, in which a classifier is used to determine which (if any) of the candidate claims are coherent and relevant for the topic. In what follows we describe these three steps in greater detail.

The Predicate Lexicon (PL) was constructed by parsing manually-detected claims (Aharoni et al., 2014) using the Watson ESG parser (McCord et al., 2012), and considering those which have exactly one verb. Then the verb and a concatenation of its right-modifiers, termed here the *predicate*, were extracted from each claim and added to the PL if they contained at least one sentiment word from the sentiment lexicon of (Hu and Liu, 2004). The sentiment criterion was added to select for predicates which have a clear stance with respect to the topic. All in all, there are 1203 entries in the PL used here.[1]

A key feature in filtering and selecting candidate claims is text similarity. The similarity between text segments was defined based on the constituent words' word2vec embedding (Mikolov et al., 2013): Consider two list of words, $l = w_1, \ldots, w_n$ and $l' = w'_1, \ldots, w'_{n'}$. Denote by $w2v(w, w')$ the word2vec similarity between w and w' - the cosine of the angle between the embeddings of w and w'. Then the similarity between l and l' is defined : $sim(l, l') = \frac{1}{n} \sum_{i=1,\ldots,n} \max_{j=1,\ldots,n'} w2v(w_i, w'_j) + \frac{1}{n'} \sum_{j=1,\ldots,n'} \max_{i=1,\ldots,n} w2v(w'_j, w_i)$ (words without embeddings are ignored). Additionally, if S is a set of text segments, define: $sim(l, S) = max_{l' \in S} sim(l, l')$.

Given a new topic t, the Generation Component sorts the predicates p in the PL according to $sim(t, p)$, and takes the top k. It then constructs k claim candidate sentences by setting the subject of the sentence to be the topic t, and the predicate to be one of these k. This may require some manipulation, as the plurality of the topic determines the appropriate surface realization of the predicate verb. We determine the topic's plurality using the Watson parser (McCord et al., 2012), and do the surface realization with SimpleNLG (Gatt and Reiter, 2009) and the NIH lexicon[2].

[1] data is avaiable at `https://www.research.ibm.com/haifa/dept/vst/mlta_data.shtml`.

[2] UMLS Reference Manual [Internet]. Bethesda (MD): National Library of Medicine (US); 2009 Sep-. 6, SPECIALIST Lexicon and Lexical Tools. Available from:

The Selection Component uses a logistic regression classifier to first predict which of the candidate claims generated by the Generation Component are valid, and then to rank the valid candidates according to the classifier's score. It receives two parameters, κ and τ. If the fraction of valid candidates (according to the classifier) is less than τ, then it selects none of them. This is designed to allow the algorithm not to synthesize claims for topics where the PL does not seem to yield a substantial number of valid claims. If the number of valid candidates is at least τ, the top κ valid candidates are returned (or all of them, if there are less than κ).

2.1 Classification Features

To describe the classification features used, we need to define - given a topic - the topic's n-gram Lexicon (n-TL). This is a list of n-grams which are presumably related to the topic. Specifically, given an n-gram, we assume its appearance in Wikipedia articles follows a hyper-geometric distribution, and estimate the distribution's parameters by counting the n-gram's appearance in a large set of Wikipedia articles. With these parameters, the p-value for its appearances in topic-related articles is calculated. The n-TL is the list of n-grams with Bonferroni-corrected p-value at most 0.05. The topic-related articles were identified manually (see (Aharoni et al., 2014)).

For a candidate claim c, denote its words by $w_1, \ldots, w_m$. Recall that c is composed of the given topic, t, and a predicate $p \in PL$. Recall also that p was extracted from a manually-detected claim c_p. Denote by t_p the topic for which c_p was detected, and by s_p the subject of the claim sentence c_p. Denote by m_t the number of words in t.

For example, consider the second candidate claim in Table 1, c = *Truth and reconciliation commissions are a source of conflict.* There t = *truth and reconciliation commissions* and p = *are a source of conflict*. p was extracted from the claim c_p = *religion is a source of conflict* in the labeled data, which is associated with the topic *atheism* (and the debatabase motion *atheism is the only way*). Hence, t_p = *atheism* and s_p = *religion.*

The classification features we used are of three types: One aims to identify predicates which are inherently amenable to generation of claims, that is, which state something fairly general about their subject, and which are not very specific to the topic in which the predicate was originally found (e.g., low $sim(p, t_p)$). The second aims to find predicates which are relevant for the new topic for which claims are synthesized (e.g., high $sim(p,\text{n-TL})$). Finally, we'd like the claim to be a valid and plausible sentence, and so look for the frequency of its words, and sub-phrases of it, in Wikipedia.

All in all 15 features were defined: m, the number of words in c; Number of Lucene hits for $w_1, \ldots, w_m$ (as a bag of words); Number of Wikipedia sentences containing all $w_1, \ldots, w_m$; Largest k, such that the k-gram $w_1 \ldots w_k$ appears in Wikipedia; Number of times the 3-gram $w_{m_t} w_{m_t+1} w_{m_t+2}$ appears in Wikipedia; Number of times p appears in a claim candidate labeled positive, and the number of times in one labeled negative (claim candidates generated for t are excluded, see Section 3 for labeling details); Inclusion of p's verb in a manually-crafted list of "causation verbs"; $sim(p,\text{n-TL})$, for $n = 1, 2, 3$; $sim(p, t)$; $sim(p, t_p)$; $sim(s_p, t_p)$; $sim(s_p, t)$.

3 Experimental Setup

We generated claims for 67 topics, extracted from debatabase motions (`http://idebate.org`) for which we have previously annotated relevant Wikipedia articles (for the benefit of the n-TLs construction; see Section 2.1). Importantly, when generating candidate claims for a topic, predicates which originated from this topic were *not used*.

For each topic 28 candidate claims were generated, and in addition one manually-detected claim (as per (Aharoni et al., 2014)) and one mock claim were included for control. The mock claim was constructed by setting the topic as the subject of a sentence, and selecting a mock predicate at random from a hand-crafted list.

These 67 × 30 candidate claims were annotated using Amazon's Mechanical Turk (AMT). In each HIT (Human Intelligence Task) we presented the annotators with a debatabase motion and 10 candidate claims, and asked which of the claims is appropriate for the motion (10 annotators per HIT).

After filtering out the less reliable annotators based on mutual agreement and control questions, a reasonable agreement was apparent (average $\kappa = 0.73$). After this filtering 45 of the initial 82 annotators remained, as well as 955 of the initial 2010 annotated candidate claims (discard-

http://www.ncbi.nlm.nih.gov/books/NBK9680/

Claim	**Original Subject**	**Label**
Democratization **contributes to stability**.	Nuclear weapons	1
Truth and reconciliation commissions **are a source of conflict**.	Religion	1
Graduated response **lacks moral legitimacy**.	The State	1
Nuclear weapons **cause lung cancer**.	Smoking	0
A global language **leads to great exhaustion**.	Great anarchy	0

Table 1: Examples of candidate claims (topics in italics, predicates in bold), the subject of the claim sentence which originated their predicate, and their label.

ing claims with less than 5 valid annotators, those without a clear majority decision, as well as the control claims). See Table 1 for some examples.

We note that annotation tasks like this are inherently subjective ((Aharoni et al., 2014) report $\kappa = 0.4$), so discarding candidates without a clear majority decision can be seen as discarding those for which the true label is not well defined. Nonetheless, the reason for discarding most of the candidate claims was annotator's (lack of) reliability, not ambiguity of the true label.

4 Experimental Results

Initially we thought to label a candidate claim as either positive or negative examples, based on the majority vote of the annotators. This lead to a seemingly 52% of the candidates being "good". However, anecdotal examination of this majority labeling suggested that the many annotators were biased toward answering "good" - even on some of the control questions which contained nonsensical sentences. This, along side relatively low mean agreement, raised the need for filtering mentioned above. After filtering, 40% of the candidate claims were taken to be positive examples. The accuracy of the Selection Component was assessed using a leave-one-out methodology, leaving out one topic at each iteration. The overall accuracy achieved by the classifier was 0.75 (Table 2 depicts the confusion matrix).

We also examined the trade-off between the number of selected candidate claims and the fraction of them which are valid. Figure 1 depicts the average precision when varying the two Selection Component parameters, κ and τ. For example, at the most conservative setting, where the component outputs at most one claim per topic, and only for a topic for which at least half the candidate claims were predicted to be valid (31 of the 67 topics), the precision is 0.94. Recall that in the entire dataset, 40% of the examples are positive.

Predict /Label	Pos	Neg
Pos	288 (30%)	145 (15%)
Neg	96 (10%)	426 (45%)

Table 2: Confusion Matrix: Number of claim candidates according to AMT annotation (x-axis) and predicted label (y-axis)

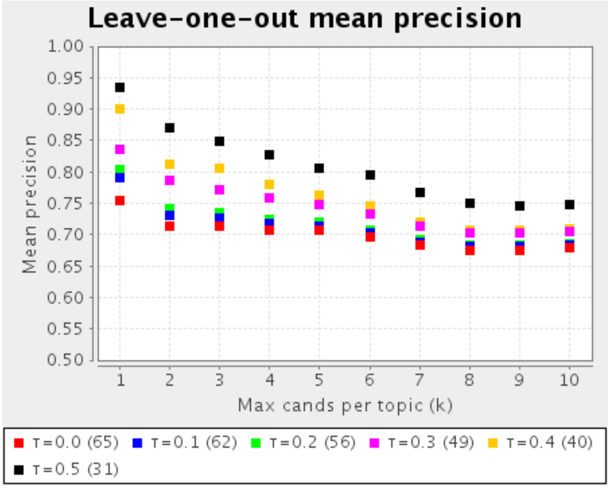

Figure 1: Mean Precision (micro average): Colors indicate different values of τ. In parenthesis is the number of topics for which claims were selected.

We note that this precision is significantly higher than reported for claim detection (Levy et al., 2014), where, for example, mean precision at 5 is 0.28 (in our case it is $0.7 - 0.8$). One should note, however, that this is not a fair comparison. First, we permit the algorithm to discard some topics. Second, here the definition of a valid claim is less strict than in (Levy et al., 2014).

Examining the impact of individual features, we first looked which of them, on their own, are most correlated with the labels. These turned out to be the number of times p appears in a claim candidate labeled positive and negative (Pearson's correlation 0.33 and -0.34 resp.). We then examined which features received the most weight in the logistic regression classifier (trained over all data; features were scaled to a $[0, 1]$). The top feature was the number of sentences in which all words appear, and following it were the aforementioned appearance counts in negative and positive examples.

5 Discussion and Work in Progress

The Generation Component can be thought of as constructing sentences by using pre-defined templates, of the form "<topic-slot> <extracted predicate>". These "generation templates" are created by "mining" a corpus of manually-detected claims and extracting the predicate from them. They are then filled in during run-time, by inserting a new topic in that slot. There are several ways which we have started exploring to extend this paradigm - automatically identifying the grammatical position of a"topic slot" in a corpus claim rather than assuming it is the subject; using unsupervised methods for mining the predicates directly from Wikipedia; and generating candidate claims by using several variants for the subject and object, rather than just the topic and the PL entry. Initial results are promising, but more work is required to achieve reasonable accuracy.

Another interesting alternative is to construct the PL manually, rather than automatically. This can be seen as analogous to Argumentation Schemes (Walton et. al, 2008). Argumentation Schemes can be thought of as templates for modeling arguments - defining a slot for a premise or two (which may be implicit), a slot for a conclusion or claim, and some fixed connecting text. While Argumentation Schemes are used for detecting (Walton, 2012) and analyzing argumentative structures, in principle they can also be used to synthesize them. In this sense, our work here can be seen as applying the same concept at finer granularity - at the claim level instead of the argument.

While at the onset we presented claim synthesis as an alternative to argumentation mining for the purpose of generating arguments, it is interesting how the two augment each other. Specifically, we have started looking at whether claim synthesis can generate claims which do not appear in our corpus (Aharoni et al., 2014), and whether matching Evidence to claims (Rinott et. al, 2015) can improve claim synthesis. Regarding the novelty of synthesized claims, we looked at 18 synthesized claims, labeled as valid for 3 topics - criminalization of blasphemy, building high-rise for housing and making physical education compulsory - and compared them to the 94 manually detected claims for these topics (each topic separately). Of the 18 claims, 5 appear to be novel.

A more circumvent method to assess novelty is as follows - for each candidate claim we looked for the most similar claim (for the same topic) in our annotated data. We then computed Pearson's correlation between these similarity scores and the labels of the candidate claim, getting a coefficient of 0.29 (p-value=10^{-27}). This is similar to the correlation between for the strongest classification features, suggesting again that many of the generated claims are not novel, yet similarity to annotated claims on its own is not enough to determine a candidate-claim's validity.

Similarly, we examined whether having a matching evidence in the annotated corpus (matches were determined using the algorithm of (Rinott et. al, 2015)), is indicative of a candidate-claim's validity. Computing correlation (over the 51 topic for which annotated evidence was available) gave a Pearson's coefficient of 0.23. This suggests that matching Evidence can be a powerful feature in improving our current classification model.

6 Acknowledgements

We thank Liat Ein-Dor for her contibution to this work, and especially for the development of the Topic Lexicons. We thank Uri Zakai, Ran Levy and Daniel Hershcovich for insightful discussions.

References

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, Noam Slonim. A Benchmark Dataset for Automatic Detection of Claims and Evidence in the Context of Controversial Topics 2014. *Workshop on Argumentation Mining, ACL*

Ngo Xuan Bach, Nguyen Le Minh, Tran Thi Oanh, and Akira Shimazu. A Two-Phase Framework for Learning Logical Structures of Paragraphs in Legal Articles. 2013. *In ACM Transactions on Asian Language Information Processing (TALIP)*. 12(1):3

Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik and Noah A. Smith. The Media Frames Corpus: Annotations of Frames Across Issues. 2015. *Association for Computational Linguistics Conference (ACL).*

Carenini, Giuseppe, and Johanna D. Moore. Generating and evaluating evaluative arguments. *Artificial Intelligence 170, no. 11*: 925-952.

A Gatt and E Reiter (2009). SimpleNLG: A realisation engine for practical applications. 2009. *Proceedings of ENLG-2009*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. 2004. *In Knowledge Discovery and Data Mining*: 168-177.

J.R. Landis and G.G. Kock. The measurement of observer agreement for categorical data". 1977. *Biometrics* 33 (1): 159174.

Irene Langkilde and Kevin Knight. Generation that exploits corpus-based statistical knowledge. 1998. *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1 (ACL '98), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA* 704-710.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni and Noam Slonim. Context Dependent Claim Detection 2014. *In The 25th International Conference on Computational Linguistics*

McCord, Michael C., J. William Murdock, and Branimir K. Boguraev. Deep parsing in Watson. 2012 *IBM Journal of Research and Development 56,* no. 3.4: 3-1.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. 2013 *In Advances in neural information processing systems,* pp. 3111-3119.

Mochales Palau, Raquel and Moens, Marie-Francine. Argumentation mining. 2011. *In Artificial Intelligence and Law*, 19(1): 1-22.

Ehud Reiter and Robert Dale. Building Natural Language Generation Systems. 2000. Cambridge University Press, New York, NY, USA.

Rinott, Ruty, Lena Dankin, Carlos Alzate, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. Show Me Your Evidencean Automatic Method for Context Dependent Evidence Detection. 2015 *In Proceedings of the 2015 Conference on Empirical Methods in NLP (EMNLP), Lisbon, Portugal,* pp. 17-21.

Stephen Toulmin. The Uses of Argument. 1958. Cambridge University Press, Cambridge.

Douglas Walton, Argument Mining by Applying Argumentation Schemes 2012. *In Studies in Logic* 4(1):38-64

Walton, Douglas, Christopher Reed, and Fabrizio Macagno. Argumentation schemes. 2008. *Cambridge University Press.*

Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. Approaches to text mining arguments from legal cases. 2010. *In Semantic processing of legal texts* 60-79.