Negating Claims

Sanjay Chhataru Gupta
Department of Computational Social Science
Paderborn University
33098 Paderborn, Germany
sanjaycg@mail.uni-paderborn.de

Abstract

This study summarizes the different methods that researchers have developed for generating negative and contrastive claims. Thus the results can be utilized in online debate systems and internet bots so that the end-user feels as if he or she is conversing with a real person.

1 Introduction

A claim that conveys an opinion on a controversial subject, either implicitly or explicitly. Over the last five years, automatic claim negation (Bilu et al., 2015; Hidey and McKeown, 2019) has gained significant attention. For example, In online debate system, the system should be capable of refuting claim and generating claims against them with a different interpretation.

The contrastive claim contains more than just viewpoint differences; it may also consist of stylistic differences, paraphrases, and other elements that differ from the original claim (Hidey and McKeown, 2019).

- Example:

! **Claim** - Narendra Modi should be Prime Minister of India.

! **Contrastive claim** - Rahul Gandhi should be Prime Minister of India.

One of the famous arguments clinic sketch from Monty Python's Flying Circus is used as an example in Computational Argumentation to illustrate how adding 'not' to a counter-argument is not always a valid counter-argument to use in conversation (Hidey and McKeown, 2019).

There are four sections in the paper. Section 2 reviews related work. Section 3 explains two different approaches to automatic claim negation, and section 4 provides some concluding remarks.

2 Related works

In recent years, negation detection has received a great deal of attention in Natural Language Processing (NLP). For instance, in the medical field, automatic negation detection is a key process in linking symptoms with diseases. Before we move on to negative sentence generation, in this section will describe how to automatically detect context dependent & independent claims and later negative sentence generation which is one of the first approaches to negation generation.

A system that detects claims automatically can be further augmented to negate them. We can use the generated claims for smooth communication. People may find it difficult to come up with an argument for discussing a particular topic. It is challenging to design a system that gives context-dependent claims.

The Figure 1 depicts the system design of automatic context-dependent claim detection (CDCD) (Levy et al., 2014).

The CDCD system was developed by using a supervised learning method that relies on label data. An annotator creates the label data by labeling the given argument as a Context-Dependent Claim (CDC) if it meets the pre-defined five criteria. A CDCD is subdivided into small parts to make the argument more focused and to avoid irrelevant text.

The CDCD system, receives input about topics and articles pertinent to the sentence component. By using sentence component, we can detect the 200 most popular sentences that contain CDCs. The output from the sentence component is passed to the boundaries component, which detects the exact boundaries of the CDC. It is then passed on to the ranking component. Sentence & boundaries scores will be input into the ranking component, and CDCs will be output as a list of 50 best candi-

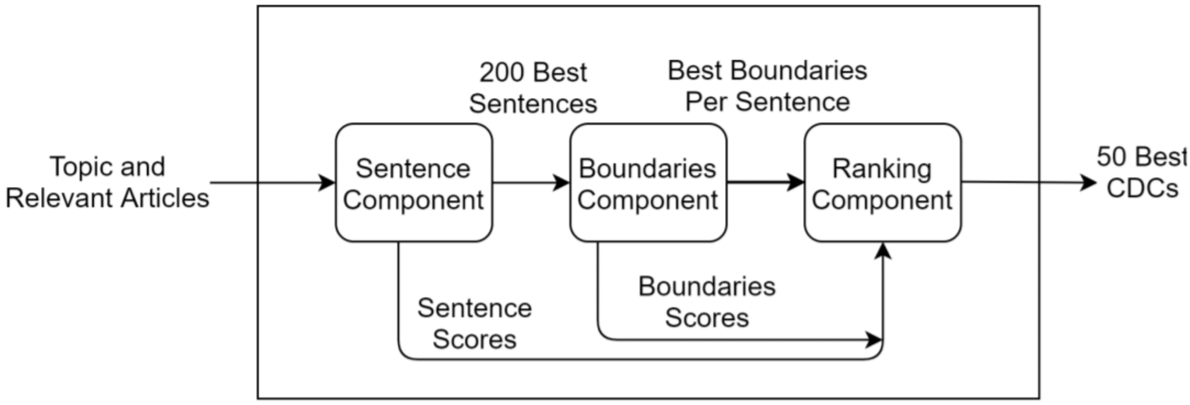

Figure 1: System design of CDCD.

dates.

The CDCD method of identifying claims may be too restrictive in some situations. Consider, for instance, extracting data from social networks containing unstructured data but many applications. Thus, we must consider a system that detects claims without regard to context. Rather than focusing on other context-dependent features such as sentiment, the Lippi and Torroni (2015) have developed system which relies on kernel methods for structured data that account for a claim's rhetorical structure.

An early approach to negating the existing sentence uses a heuristic approach to create a negative sentence by Ahmed and ip Lin (2014). Ahmed and Lin developed a system that will automatically generate all possible negation sentences from input statements. In the end, they have to use a predefined semantic heuristic to create a more user-friendly version of the sentence.

The Figure 2 demonstrate the workflow of the Negative sentence generation system. There are two parts to the system: the first part detects all possible scopes of statements; the second part generates statements by using the 'NOT' on the negative parts.

As part of the first step, the system will call a Part-of-speech (PoS) tagger to label each word in the sentence. A parse tree is then generated after parsing the statement. After that, apply predefined heuristics to the parse tree to identify each sentence's scope. A program outputs negations based on each scope once the scope has been determined. In the end, it turns out scope is of great importance to negation. This system has the limitation that it does not work well with compound sentences.

3 Negative Claim Generation

Having already studied claim detections and negative sentence generation in the preceding section 2, we will employ these concepts as a foundation for negating claims.

After claim detection, we can augment the claim to negation, and use this negated claim in conversation to refute the original claim. As an example, we can use negated claims as a way to refute or defend the original claims in an online debate system. The objective of this section is to examine two state-of-the-art approach to negate claims.

3.1 Automatic Claim Negation

In this approach, the main goal is to generate negated claims and use those to generate plausible claims in an online debate-support system. Bilu et al. (2015) use rule-based and statistical approach to solve the problems, respectively. A claim negation can be generated automatically by using four levels of complexity according to the Bilu et al. (2015) team.

1. Grammer - The generated claim should be grammatically correct.

   - Example:
   ! **Claim** - All information should be freely distributed and unrestricted.
   ! **Negated Claim** - All information should **should not** be freely distributed and unrestricted.

   Since the negated claim is incorrect, but this is a rare case that could be easily fixed.

2. Clarity - The generated claim should be grammatically correct, but unclear and incoherent.

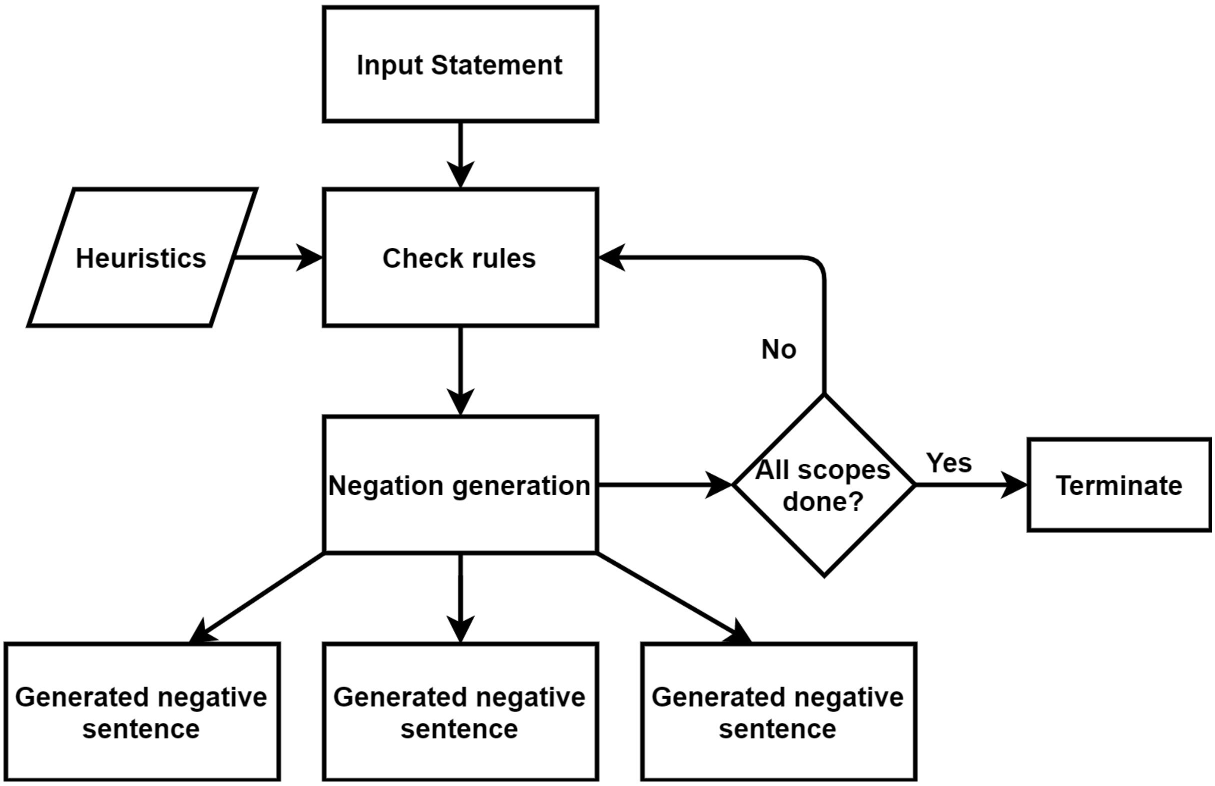

Figure 2: Workflow of the Negative sentence generation system.

- Example:
  - **Claim** - School should be made to fit the child, rather than the other way around.
  - **Negated Claim** - School should **not** be made to fit the child, rather than the other way around.

  Despite being grammatically correct, the above-negated claim is unclear. This implies that a child shouldn't be forced into school since it's very meaningless.

3. Opposition - Despite being grammatically correct and clear, the generated claim does not express the opposite of the original claim.

   - Example:
     - **Claim** - Children who fail to engage in regular physical activity are at greater risk of obesity.
     - **Negated Claim** - Children who **do not** fail to engage in regular physical activity are at greater risk of obesity.

   In the above negating claims, the opposite is not stated for the original claim, therefore it is not valid negation.

4. Usability - In spite of satisfying all three above criteria, the generated claim might still not be plausible to use in a discussion.

   - Example:
     - **Claim** - The selection process should not be based on some arbitrary or irrelevant criterion.
     - **Negated Claim** - The selection process **should** be based on some arbitrary or irrelevant criterion.

   The above-negated claims satisfy the above-all claim, but it is difficult to imagine someone stating that in conversation.

3.1.1 Algorithm

Figure 3, depicts automatic claim negation system architecture (Bilu et al., 2015). It is divided into two parts; first stage is rule-based algorithm which will generate the claim negation, and in second stage, classification scheme is use to check whether it clear the Usability criteria of complexity.

To better understand the where the most of challenges lies Bilu et al. (2015) team did the preliminary analysis. Five annotators given task to manually determine the difficulty level of first 200 claims in the dataset published in (Aharoni et al., 2014).

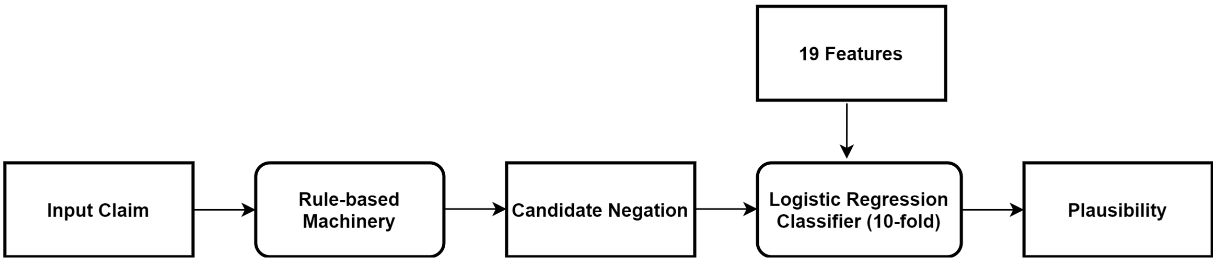

Figure 3: System design of Automatic claim negation.

The difficulty levels are divided into three labels. If annotator able to negate claims just by adding or removing **no, not, does not,** or **do not** in original claim is called 'Type 1'. 'Type 2', if annotators able to think of negation of claim but not through simple modification (e.g.- 'Type 1' label). 'Type 3', if annotators not easily able to phrase a clear negation. The result of the annotation, about 75% of the claims that can be negated by simple rules (Type 1). Around 23.5% negation can be use in conversation. As a result suggest, main challenges lie in determining the usability of negated claim.

The first part of algorithm take input as claim and use rule-based machinery to generate negation. Rule-based machinery is based on set of few pre-defined rules. Tokenize the input claim and label the tokens for part of speech, insert or remove the does not or do not.

The output of first stage is given input to second part of th algorithm. In the second part we are focusing on to determine the usability of negated claims. Based on preliminary analysis, Bilu et al. (2015) team created set of 19 features. The features include counts, tokens, PoS Tags, Sentiment, frequency. Each candidate negation is transformed into 19-dimensional feature vector. The logistic regression classifier are trained and tested based on feature representations in a 10-fold cross validation framework.

3.1.2 Results

Bilu et al. (2015) team tested their proposed system architecture on 1240 claims dataset published in (Aharoni et al., 2014). This claim is given input to first stage algorithm, which will generate the 1,240 pairs of the form (Claim, candidate negation). This output pairs are given to five annotators to label the negated claims according to four complexity criteria mentioned in section 3.1.

In the second stage, all negated claims are transformed into 19-dimensional feature vector. This features are used to train and test the Logistic regression classifier. The outcome of classifier is better than the annotators results. Around 96% of claim is labeled as correct for Grammar and approximate 80% of claim is labeled as Clarity and Opposition. Only 50% of negated claim can be usable for conversion which is double as per their preliminary study. The limitation of this proposed system is that it explicitly negate the claims.

3.2 Fixed That For You (FTFY)

The FTFY, is a labeled reddit corpus of comment-pairs published by Hidey and McKeown (2019) team. Christopher and Kathy, train the neural networks on this corpus to update the original claim and generate new claim with a different view. In order to generate claim with different view required understanding of argument context which we already study in section 2.

3.2.1 Data

To generate contrastive claims we need large amount of datasets which reflects this phenomenon. Training data is collected through the scraping the social media site Reddit for comments containing acronym *FTFY*[1]. FTFY responses are used to respond to another comment by editing the part of parent comments. As result, around 2,200,258 pairs of historical data are scrapped from Reddit. The pair consists of a parent and an FTFY. Comment-pairs are then train on binary classifier to predict the contrastive claims. The classifier filter out not required comment-pairs and return around 108 Billions contrastive claims. The 1,083,797 pairs of contrastive claims are divided into about 1% for development and 0.6% for test and around 98.4% for training the neural models.

3.2.2 Algorithm

The Figure 4 shows the workflow of the algorithm developed by Hidey and McKeown (2019). The goal is find the words in original claims that should

[1] https://www.reddit.com/r/ftfy/

be remove or replace and generating appropriate suggestions on that context.

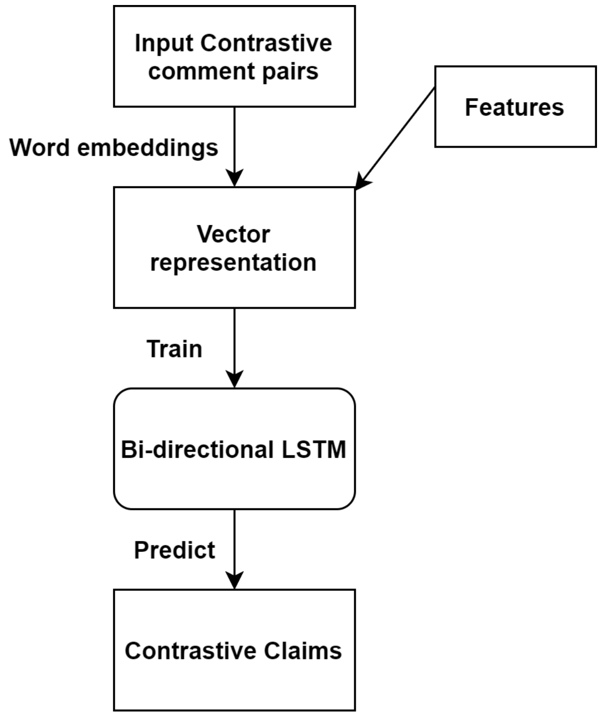

Figure 4: Work flow of Contrastive claims generation algorithm.

Hidey and McKeown (2019) team use neural sequence-to-sequence encoder-decoder models (Sutskever et al., 2014) with attention for experiments. The input contrastive claims are converted to 300-dimensional vectors representation. Using GloVe (Pennington et al., 2014) word embedding is features is calculated for words in the entire comment span and features from the aligned phrases span only. The vector representation is pass to neural sequence-to-sequence encoder-decoder models (Sutskever et al., 2014). The output of the model is contrastive claims.

The below examples are the output of Neural model (Hidey and McKeown, 2019). In the first example the contradiction is due to the choice of operating system. The second example is from politics category and contrast between allowing markets to regulate themselves versus an increased role of government.

- Example:

! **Parent** - i know that this is an unofficial mod, but xp is the best os for this machine

! **Model** - linux is the best os for this machine

- Example:

! **Parent** - ah yes the wonders of the free market

! **Model** - ah yes the wonders of government intervention

The BLEU (Papineni et al., 2002) score was 47.28, which shows the difficulty of automatic evaluation.

4 Conclusion

In the course of this study, it became clear that the automatic claim negation system was rather naive. Over one million pairs of contrastive claims are available. The proposed approach 3.2, can be incorporated into Online debate system.

Acknowledgments

I sincerely thank Noam Slonim and his team at IBM for sharing datasets. I also thank Jun.-Prof. Dr. Henning Wachsmuth and Milad Alshomary for their helpful comments on presentation.

References

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics, ArgMining@ACL 2014, June 26, 2014, Baltimore, Maryland, USA*, pages 64–68. The Association for Computer Linguistics.

Afroza Ahmed and King ip Lin. 2014. Negative sentence generation by using semantic heuristics. In *The 52nd Annual ACM Southeast Conference (ACMSE 2014), March 28-29, Kennesaw, GA, USA*. The Association for Computing Machinery.

Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. 2015. Automatic claim negation: Why, how and when. In *Proceedings of the 2nd Workshop on Argumentation Mining, ArgMining@HLT-NAACL 2015, June 4, 2015, Denver, Colorado, USA*, pages 84–93. The Association for Computational Linguistics.

Christopher Hidey and Kathy McKeown. 2019. Fixed that for you: Generating contrastive claims with semantic edits. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1756–1767. Association for Computational Linguistics.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1489–1500. ACL.

Marco Lippi and Paolo Torroni. 2015. Context-independent claim detection for argument mining. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 185–191. AAAI Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.