Negative Sentence Generation by Using Semantic Heuristics

Afroza Ahmed, King-Ip Lin
Department of Computer Science
The University of Memphis
Memphis, TN 38152, USA.
davidlin@memphis.edu

ABSTRACT
This research is focused on finding all possible interpretations of negative sentence from a given sentence. The challenge exists where we could possibly match up with the human interpretation of negative sentence semantically through this automatic negation generation. This paper studies natural language generation of negative sentences and the problem of how to handle the different emphasis of words that occur in the generation processes. The main concern is to find all the possible candidates of negation which can lead to negative sentence generation. To have semantically correct negation we apply some fixed heuristics. Hence, this paper studies about automatic generation of all possible versions of negation by applying heuristics to produce more human friendly natural language versions. We evaluated the system by using sentences from Manually Annotated Sub-Corpus (MASC).

Keywords
Candidate scope, Negation generation, POS-Tag, Scope prediction.

1. INTRODUCTION

Negation processing has received significant attention over the past few years. For instance, automatic negation detection is a very important topic in bioinformatics, where detecting negation is crucial in, for instance, reaching the right diagnosis.

In this paper we look at a slight different, but yet as significant problem, -- negation generation. That is, given a statement, we would like to automatically generate its negation. We believe this is useful in many applications. For example, with the rise of conversational agents, in order to generate smooth and adaptive conversation, one may want to negate a certain fact by either the human speaker interacting with the agent, or a given fact in the agent's "knowledge base". Ability to automatic negate a statement is a crucial step towards generating the smooth conversation.

Another potential application is locating opinion over the web. For instance, currently keyword search is the most common way of extracting information from the web. Suppose a company want to find information to contradict an existing claim. Currently one can do a keyword search on the claim and hope the search return the contradictory information at the top, and the corresponding negation algorithm detects it. However, an alternative is to automatically negating the claim, then search the web to find the statement that contains the negation. This enables such negative information to be retrieved much easily.

The 52nd Annual ACM Southeast Conference (ACMSE 2014), March 28-29, Kennesaw, GA, USA.

Thus we believe there is tremendous value in automatic negation generation. While the task sounds simple, there are many challenges, including the following:

- What to negate. Even for a simple sentence there are many things that can be negated. For example, a simple statement such as "John eats candy" can be negated in 3 ways (in which each word can be negated).
- How to negate. One need to generate negations that are both grammatical and colloquial.
- Conditionals and logical quantifiers. For instance, statements with "if .. . then", "always", "sometimes" denote some logical meaning that care must be taken in negating them. On the other hand even though the negations can be done correctly logically, it may not be the most colloquial way of expressing it.

In this paper, we describe an algorithm that we built by applying some standard rules in negating statements. We use both syntactic analysis of the statement itself, and semantic analysis (using external sources such as Wordnet). We are able provide good negations to many statements. We evaluated our methods and show that the results are promising.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 provides the details of our algorithm. We have some experimental results at section 4, and section 5 provide some concluding remarks and directions for future work.

2. RELATED WORK

While there were quite a bit of work related with negation detection and negative sentence analysis, significant less work are done for negative sentence generation. Negation detection has received a lot of attention in the medical field as it helps system to determine relationships (or lack thereof) between symptoms and dieseases.

Valerio Basile used the NLP toolchain that is used to construct the Groningen Meaning Bank to address the task of detecting negation cue and scope, as defined in [7]. The toolchain applied the C&C tools for parsing, using the formalism of Combinatory Categorical Grammar, and applied Boxer to produce semantic representations in the form of Discourse Representation Structures (DRSs). For negation cue detection, the DRSs were converted to flat, non-recursive structures, called Discourse Representation Graphs (DRGs).

In Negation detection using machine learning by James Paul White cue detection was performed using regular expression rules extracted from the training data. Both scope tokens and negated event tokens were resolved using a Conditional Random Field (CRF) sequence tagger.

Recognition of phrases containing negation, particularly in the medical domain, using regular expressions has been described using several different approaches. Systems such as Negfinder [19] and NegEx [20] use manually constructed rules to extract phrases from text and classify them as to whether they contain an expression of negation.

UCM-2 by [21] made use of an algorithm that detected negation cues, like no, not or nothing, and the words affected by them by traversing Minipar dependency structures. The scope of these negation cues was also computed by using a post processing rule based approach that took into account the information provided by the first algorithm and simple linguistic clause boundaries. However, the initial version of this system could only handle the annotations of the Bioscope corpus [9].

In FBK [8] a system was developed that exploited phrasal and contextual clues apart from various token specific features. In CRF for resolving scope of negation [1] the system was designed to detect negation in English text. They addressed three tasks: negation cue detection, negation scope resolution and negated event identification. They trained their (CRF) model on lexical, structural, and syntactic features extracted from labeled data. The system detected negation cues with 90.98% [1].

In UABCoRAL[3] they proposed similar system to [1], but their system used lexicon signals to detect negation cues.
Then, they applied machine learning approaches to detect the scope and negated event for each negation cue identified in the first phase.

3. NEGATION GENERATION -- OUR APPROACH

In this section we present our algorithm for negation generation.

3.1 Problem definition and challenges

The goal of our algorithm is to generate the negation of a given statement. This problem can be divided into two parts: what to negation (scope detection), and how to negate.

What to negate (scope detection)

For a given sentence there are many potential way to negate it. Taking an example from [2], the straightforward negation of the sentence *He often flies to beautiful Paris* is *He doesn't often fly to Paris*. However, if we read the sentence with different emphases, then we found:

I. *He* doesn't often fly to beautiful Paris.
II. He doesn't often *fly* to beautiful Paris.
III. He doesn't often fly to beautiful *Paris*.
IV. He doesn't *often* fly to beautiful Paris.
V. He doesn't often fly *to beautiful* Paris.

This is not clear that (I) entails that others do fly to Paris often, that (II) entails that he usually goes Paris by boat or train, or (III) entails that he often flies to places other than Paris and that (IV) entails that he flies to Paris rarely, or even (V) that maintain Paris is not beautiful. So in natural language when it is come to negating a sentence, several possible scenarios can be valid. We denote each possibility as the *negative scope* of a statement. So if these entailments hold, then (I) cannot be always the only negative of original true sentence.

There is research on picking the most likely scope (denoted as focus) of a sentence to negate. However, in our case we focus on generating the negation, so our system needs to take into account all possible scopes. So in this case our algorithm detects all possible scopes of a statement in the first place. This means our algorithm can generate multiple negations for one input.

How to negate (generating the statements)

The most common way to generate the negation is by using NOT on the negative parts (we denote it as the NOT hypothesis). For instance, with the statement above, a mechanical way of generating negation is by generating the following 5 statements:

1. (Not He) often flies to beautiful Paris
2. He (not often) flies to beautiful Paris
3. He often (not flies) to beautiful Paris
4. He often flies to beautiful (not Paris)
5. He often flies to (not beautiful) Paris

According to the NOT hypothesis, NOT modifies the different parts of the sentence to give all possible negations having the scenarios as (Not he), (Not fly), (Not Paris), and (Not often) and (Not beautiful) [2].

While this seems straightforward enough, the sentences that are generated are likely to be awkward at best, completely unreadable at worst. Thus one needs to modify this simple approach in order to generate "natural" negative sentences.

3.2 Our approach

Our approach can be divided into 4 major steps for negation generation.

1. Part of speech tagging
2. Predicting the possible scopes/candidate selection
3. Negating the statements based on scopes
4. Use synsets to provide human friendly negation

Figure 1 shows the flowchart of the basic work flow of the system.

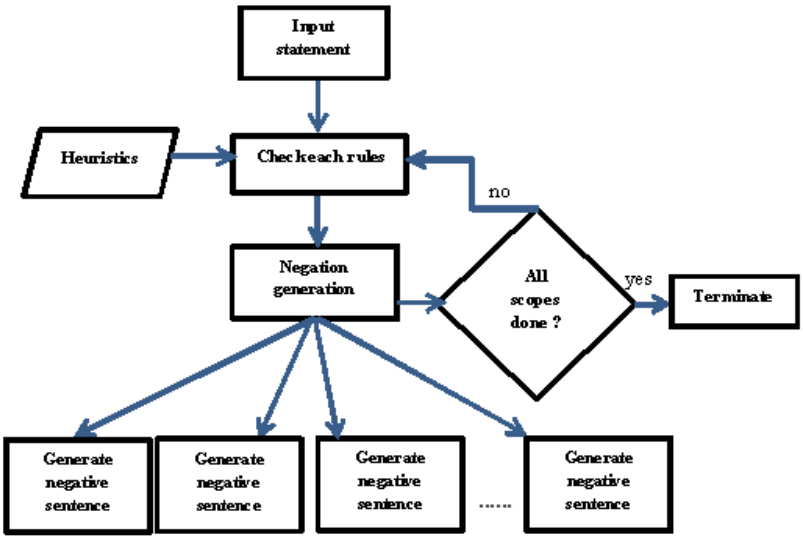

Figure 1. Work flow of the proposed system.

Part of Speech tagging and Scope detection

In our approach, we use the part-of-speech of the words to provide us with hints about the location of the scopes. We used the MaxEnt Model with TAG dictionary from the open NLP parser library for part-of-speech tagging. Using this model, once the system accepts the input, it invokes the Part of speech (POS) tagger to tag each term in the sentence. For example, if the input sentence is "People continue to inquire the reason for the race for outer space", the output sentence provided by the tagging functions is "People/NNS continue/VBP to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN".

Once tagging is done, we parse the statement to create a parse tree. Then we apply a set of heuristics to locate the various possible scopes for each sentence. The heuristics are created based on examining a large set of sentences and their potential negations. The heuristics are as follows:

1. If a sentence starts with Noun Phrase (NP) and NP is followed by a verb phrase (VP) then the NP is a scope. Example: : [The cow] NP [eats grass]VP. So "The cow" is a scope.
2. Adjectives of the sentence. Example: She is [good]Adj at math. "Good" here is a scope
3. If there is any Prepositional phrase (PP) after a verb (VB), then the PP is scope. Example: [The baby]NP [eats rice]VP [with spoon]PP. "With spoon" is a scope.
4. An adverbial phrase after a verb. Example: She went to meeting [on Monday]. "on Monday" is a scope.
5. Verbs and verb arguments (NP that are children of the verb phrase in the parse tree). Example: She [ate]VB chips and biscuits. Scopes: "ate", "chips", "biscuits"
6. Pre-determiners. Example: All the people left the place. All (the pre-determiner) is a scope.

Using these heuristics, our algorithms can determine a set of scopes that are passed to the next phase for negation generation.

Negation generation (and making it human friendly)

Once the program determines the scope(s) available, it then pass each scope to the algorithm to generate the negations. It turns out that the part of speech of the scope that is being negated plays the primary role on how the sentence should be negated. We distinguish the following cases:

- Verbs: This is the simplest case. Typically adding "not" (more precisely "do not" or "does not")
- is the correct action. One need to careful to include the correct auxiliary verb, as well as tenses and third person singular forms.
- Adverbs and adjectives: Most adverbs and adjective has a set of antonyms. Typically replacing an appropriate antonym for the corresponding adjective and/or adverbs will do the job. Here we use Wordnet that provides an interface for one to search for antonyms of adjectives and adverbs. If no antonym is available, we use the NOT hypothesis and add NOT before the adjective and/or adverb.
- Pre-determiners. There is only a small number of pre-determiners, and one can relatively easily hard-code the required negation for them (e.g. "all" to "not all", "either" to "neither" etc.).
- Nouns. This can appear in noun phrases and prepositional phrases. This is the trickiest case, where the NOT hypothesis seldom (if at all) works. For instance "not Paris", "not the cow" etc. is not grammatical. Furthermore, the way to negate the noun depends on the type of the noun. For instance, "not Paris" should translate to "somewhere other than Paris", where "not the cow" should negate (possibly) to "something other than the cow".
  In our implementation we overcome this by using hypernyms (from Wordnet). We first locate the noun in Wordnet, and then recursively traverse up the hypernym hierarchy. As soon as a level that denotes the item is a thing, place, or person, then we use the

appropriate term (something, somewhere, someone etc.) to form the negation sentence.

We repeat the process for each scope located, and form one negation sentence per scope. Notice that at this stage we do not combine multiple scopes for a sentence for fear of having double negatives.

3.3 Examples

The following are a couple of examples on how our algorithm works.

Example 1: The cow eats grass with spoon.

After tagging: The_DT cow_NN eats_VBZ grass_NN with_IN spoon._NN

Scope verb (eats): Negated Statement:: The cow *doesn't* eat grass with spoon

Scope PP after verb (with spoon): Negated Statement: The cow eats grass with *something other than spoon*

Scope NP (The cow): Negated Statement. *Something other than the cow* eats grass with spoon

Example 2: He often files to London

After tagging He_PRP often_RB flies_VBZ to_TO London._NNP

Scope adverb (often): Negated statement: He *rarely* flies to London.

Scope abverb (often): Negated statement: He *infrequently* flies to London.

Scope verb (fly): Negated statement: He often *doesn't* fly to London

Scope noun pharse (He): Negated statement: *Someone other than him* often flies to London

Scope PP (to London) Negated statement: He often flies to *somewhere other than London*

Limitations

Our system generally works well for simple sentences. However, in case of compound sentences, especially one that with causal relationships, our heuristics can run awry. For example:

Example 3: It is in fact the principal island of an archipelago of 14, most of them little more than piles of rocks.

If we pick the first adverb (most) as the scope, the negated statement will be: “It is in fact the principal island of an archipelago of 14, *least* of them little more than piles of rocks.”, which does not make much sense. It should be noted that by negation the first verb (is), we still get a reasonable negation: “It *is not* in fact the principal island of an archipelago of 14, most of them little more than piles of rocks.

Thus we need to evaluate our performance to see how good our algorithm is doing its job. This will be done in section 4.

3.4 Implementation details

The tool has been implemented in Core Java.. The third party libraries, which have been imported for this tool, are OpenNLP, Wordnet and RiTa Wordnet and nltk for python.

OpenNLP is used for tagging and parsing for scope detection purpose. In order to access the program

Wordnet is a lexical database for the English language [13]. It groups English words into sets of synonyms called *synsets*, provides short, general definitions, and records the various semantic relations between these synonym sets. As mentioned above, Wordnet provides synsets information to enable us to find antonyms and hypernyms.

To incorporate Wordnet into our Java program, we use RiTa Wordnet, developed by Daniel C Howe, that provides simple access to the WordNet ontology for language oriented artists [14]. RiTa Wordnet provides a core.jar file, which needs to be imported into the classpath for the Java program. Once this file is imported, we can use RiTa WordNet’s functions to access WordNet dictionary to retrieve information. To get the antonym, we need two things. First we need the word of which the adjective needs to be found. And then we need the best position of the word in the sentence. This helps WordNet give as much accurate antonyms as possible.

A separate python script is called from java which is written to check the word’s synset classification. For this, we used nltk (natural language toolkit) for Wordnet. The method hypernym_paths(self) is used which will get the pathS from this synset to the root, where each path is a list of the synset nodes traversed on the way to the root. It will return a list of lists, where each list gives the node sequence connecting the initial Synset node and a root node. For example, Synset hypernym paths for “car” is:

[Synset('entity.n.01'), Synset('physical_entity.n.01'), Synset('object.n.01'), Synset('whole.n.02'), Synset('artifact.n.01'), Synset('instrumentality.n.03'), Synset('container.n.01'), Synset('wheeled_vehicle.n.01'), Synset('self-propelled_vehicle.n.01'), Synset('motor_vehicle.n.01'), Synset('car.n.01')]

From the above example we can see that we easily find out the classification details of a word through hypernym_paths. This helps us to apply more human friendly interpretation of the negation.

4. EXPERIMENTAL RESULTS

In this section we presented our experimental results. As stated previously, while our system works pretty well with many cases, it is by no means perfect, and here we present our evaluation based on testing on a test corpus.

For our experiment, we take sentences from the MASC (Manually Annotated Sub-Corpus) [12] for usages. MASC is distributed without license or other restrictions from the

American National Corpus website. It is also freely available from the Linguistic Data Consortium (LDC). MASC provides sentences that are annotated with part-of-speech and other information such as name entities (e.g. time, location etc.) that is very useful to help us evaluate to effectiveness of our method, especially with nouns.

We ran 20 different types of statements including simple and complex statements from the corpus. We manually check for possible scopes for each statement. Then we check how many possible scopes it could detect automatically. Next we check on how many correct negative statements were generated out of those scopes. Here the findings are submitted in the form of a table in Table 1 and Table 2. In Table 1 we have the scope-wise findings based on how many each scope was corrected detected. From the table we can see that scope (scope 1) regarding VB has the highest success ratio (95%). Hence, we need to concentrate more on other scope's scenarios to improve their result.

Table 1. Evaluation Results from 20 Sample runs

Scope	Correctly detected	Total Scopes	Incorrect	Success Ratio
1	**19**	**20**	**1**	**95%**
2	**5**	**7**	**2**	**71%**
3	**13**	**16**	**3**	**81%**
4	**9**	**14**	**5**	**64%**
5	**5**	**5**	**10**	**50%**
6	**2**	**4**	**2**	**50%**

We also examined the total number of scope detected as compared with what should be detected and we see that 80% of the time, the required scope(s) are detected.

5. CONCLUSION

In this paper, we developed a tool to automatically generate all possible interpretations of negation from a given input statement. We used some heuristics to get outputs with more human friendly form. Our methods first find all possible candidates for negation scopes and then generates negative sentence depending on the particular scope logic. We tried to achieve negation generation in semantic representation of natural language. We illustrate the importance of predicting scopes to generate humanly possible negation interpretations of a given statement.

Although this work gave us a solid starting point for this area there is more ways to enhance it to perfection. We are yet to evaluate this tool with much broader corpus and we want to implement more linguistics aspects so that more human friendly sentences could be generated.

6. REFERENCES

1] Amjad Abu-Jbara and Dragomir Radev. 2012. UMichigan: a conditional random field model for resolving the scope of negation. *In Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, Stroudsburg, PA, USA, 328-334.

2] Barnes, Jonathan. 1995. Formal language and natural language: Negation in English. *Rue Descartes* 14 , 153-184. Presses Universitaires de France, France.

3] Binod Gyawali and Thamar Solorio. 2012. UABCoRAL: a preliminary study for resolving the scope of negation. *In Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, Stroudsburg, PA, USA, 275-281.

4] Blanco, Eduardo, and Dan I. Moldovan. 2011. Some Issues on Detecting Negation from Text. FLAIRS Conference.

5] Eduardo Blanco and Dan Moldovan. 2011. Semantic representation of negation using focus detection. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Stroudsburg, PA, USA, 581-589.

6] Junhui Li, Guodong Zhou, Hongling Wang, and Qiaoming Zhu. 2010. Learning the scope of negation via shallow semantic parsing. In Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 671-679.

7] Lifeng Jia, Clement Yu, and Weiyi Meng. 2009. The effect of negation on sentiment analysis and retrieval effectiveness. *In* Proceedings of the *18th ACM conference on Information and knowledge management* (CIKM '09). ACM, New York, NY, USA, 1827-1830.

8] Md. Faisal Mahbub Chowdhury. 2012. FBK: exploiting phrasal and contextual clues for negation scope detection. *In Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, Stroudsburg, PA, USA, 340-346.

9] Miguel Ballesteros, Alberto Díaz, Virginia Francisco, Pablo Gervás, Jorge Carrillo de Albornoz, and Laura Plaza. 2012. UCM-2: a rule-based approach to infer the scope of negation via dependency parsing. *In Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, Stroudsburg, PA, USA, 288-293.

10] Morante, Roser. 2010. Descriptive analysis of negation cues in biomedical texts .*In Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 1429–1436, Valletta.

11] Partha Pakray, Pinaki Bhaskar, Somnath Banerjee, Sivaji Bandyopadhyay, Alexander F. Gelbukh. 2012. An Automatic System for Modality and Negation Detection. *In proceedings of CLEF (Online Working Notes/Labs/Workshop).*

12] Passonneau, R., Baker, C., Fellbaum, C., Ide, N. 2012. The MASC Word Sense Sentence Corpus. *Proceedings of the Eighth Language Resources and Evaluation Conference*, Istanbul. http://www.anc.org/data/masc/wordnet-framenet-annotations/

13] Princeton University "About WordNet." WordNet. Princeton University. 2010. http://wordnet.princeton.edu [22nd November 2013]

14] RiTa WordNet :- http://www.rednoise.org/rita/wordnet/documentation/riwordnet_class_riwordnet.htm [22nd November 2013]

15] Roser Morante and Eduardo Blanco. 2012. *SEM 2012 shared task: resolving the scope and focus of negation. *In*

Proceedings of the First Joint Conference on Lexical and Computational. Association for Computational Linguistics, Stroudsburg, PA, USA, 265-274.

16] Roser Morante, Sarah Schrauwen, and Walter Daelemans. 2011. Corpus-based approaches to processing the scope of negation cues: an evaluation of the state of the art. *In Proceedings of the Ninth International Conference on Computational Semantics* (IWCS '11). Association for Computational Linguistics, Stroudsburg, PA, USA, 350-354.

17] Wiegand, Michael, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. 2010. A survey on the role of negation in sentiment analysis. *In Proceedings of the workshop on negation and speculation in natural language processing*, pp. 60-68. Association for Computational Linguistics.

18] Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. UGroningen: negation detection with discourse representation structures. *In Proceedings of the First Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*, Stroudsburg, PA, USA, 301-309.

19] Mutalik PG, Deshpande A, Nadkarni PM. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. J Am Med Inform Assoc. 2001 Nov-Dec;8(6):598-609.

20] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform. 2001;34:301-10.

21] Ballesteros M, Díaz A, Francisco V, Gervás P, Carrillo de Albornoz J, Plaza L. 2012. UCM-2: a Rule-Based Approach to Infer the Scope of Negation via Dependency Parsing. *SEM Shared Task 2012. Resolving the Scope and Focus of Negation.