Calculating Folktale Similarity using Frames and Agents

Final Project Report

LIN 350 – Computational Semantics

VICTORIA ANUGRAH LESTARI (VAL565)

1. Introduction

Folktales are stories, usually written by unknown authors, that are passed on from generations to generations both in oral and written form. Sometimes folktales were made up to explain scientific conditions that were not understood at that time, such as the forming of a mountain or a river. Thus, they are a valuable resource to learn about various cultural heritages. In computational linguistics, there are many researches on the topic of folktales [1] [2] [3] [6] [7], even though not as many as researches done on news articles and social media texts.

Usually, researchers perform experiments to discover the topic of folktales [1] [3], or to calculate the similarity of several folktales [6]. Researches on folktales are interesting because the words used in the folktales are different from those in news articles. The tools that can parse sentences and extract words correctly in news articles sometimes make mistakes when doing so on a corpus of folktales. Thus, the techniques used for preprocessing folktales should be a little different from those used for preprocessing news articles. Gold standards can be obtained through manual annotation or using literature guides, such as Vladimir Propp's Morphology of the Folk Tale [9] or Aarne-Thompson-Uther classification systems [4].

In this project, I want to calculate the similarity between folktales, measured by two components: the similarity of order of events in the folktales, and the similarity of the actors and patients that are involved in the events. For example, the story "The Wild Swans" by Hans Christian Andersen and the story "The Six Swans" by Brothers Grimm are highly similar. The differences are that Andersen's version has eleven swans, while Grimm's version has six swans, and that the antagonist of Andersen's version is the Archbishop, while the antagonist of Grimm's version is the King's mother. A less similar version is Brother Grimm's "The Twelve Brothers", in which the brothers turned into ravens instead of swans, and the princess did not make shirts out of nettles for her brothers.

I have conducted a similar research for my undergraduate thesis. However, the methods I choose for this project will be different. In the previous work, a story is represented by a graph of actors and events [8], and the graphs are compared to yield a similarity value between 0 and 1. In this research, I do not construct a graph to represent a story. Instead, a story is represented by a vector of frames that are extracted by automatic semantic role labeling. By using frames, I intend to capture the generality of a story plot instead of comparing word-per-word as I did in my previous research.

2. Research Questions

The questions I want to ask regarding this projects are:

- How can we model intuitive ideas about similarity between folk tales?

- Is similarity about similarity in protagonists, similarity in events, or both?
- In what level the similarity be calculated? Word by word, paragraph by paragraph, topic by topic, or concept by concept (more abstract)?
- Should the folktales be collected in groups first (using Latent Dirichlet Allocation/LDA or other method) and the comparison is done inside a group?
- How to determine the performance of the method conducted in this research? What gold standards must be used? How to construct the gold standard?

3. Data

I am using over 350 folktales from Andrew Lang's Fairy Tales Book, downloaded from Project Gutenberg. I have collected the raw text (.txt files) when I did my undergraduate research. Each folktale's length ranges from 300 to 10,000 words.

4. Method

The steps I did in this project are preprocessing the texts, identifying the agents and the frames, and comparing the similarity results with those from LDA. The tools used in this project are Natural Language Toolkit (NLTK)[1], gensim[2], and SEMAFOR[3].

4.1 Preprocessing

Initially, I had 351 files, each containing a folktale. In a file, each paragraph is divided with a newline character. Since SEMAFOR accepts the formal where each line contains a sentence, I used NLTK sentence tokenizer to identify sentences and adjust the file formatting to suit SEMAFOR.

4.2 Similarity between frames

According to the FrameNet website[4], the basic idea of frame semantics is that "the meanings of most words can best be understood on the basis of a semantic frame: a description of a type of event, relation, or entity and the participants in it." In other words, a frame is a description of an event, relation, or entity and its relationship with its participants. For example, the concept "cooking" has the participants of someone who cooks (person), something to be cooked (food), something to cook the food (cooking utensils), and the source of heat (stove, microwave, etc.).

4.2.1 Identifying the frames

After the preprocessing performed in step 4.1, the files were ready to be processed further by SEMAFOR. Since SEMAFOR could not run on my Windows computer, I accessed UTCS' Linux server (linux.cs.utexas.edu) via remote server (SSH) to run SEMAFOR on my files. I wrote a Python script to run SEMAFOR iteratively using subprocess. The output for each file was an XML document containing the frames of the story. Each sentence was annotated into frames.

Here is a part of an XML file produced by SEMAFOR. The sentence is "Once upon a time there lived a fairy whose name was Dindonette."

[1]http://www.nltk.org/
[2]https://radimrehurek.com/gensim/
[3]http://www.cs.cmu.edu/~ark/SEMAFOR/
[4]https://framenet.icsi.berkeley.edu/fndrupal/

```
<sentence ID="0">
        <text>Once upon a time there lived a fairy whose name was Dindonette .</text>
        <annotationSets>
         <annotationSet ID="0" frameName="Residence">
          <layers>
           <layer ID="1" name="Target">
            <labels>
             <label ID="101" end="27" name="Target" start="23"/>
            </labels>
           </layer>
           <layer ID="2" name="FE">
            <labels>
             <label ID="201" end="21" name="Location" start="17"/>
             <label ID="202" end="15" name="Resident" start="10"/>
            </labels>
           </layer>
          </layers>
         </annotationSet>
         <annotationSet ID="1" frameName="Being_named">
          <layers>
           <layer ID="101" name="Target">
            <labels>
             <label ID="10101" end="46" name="Target" start="43"/>
            </labels>
           </layer>
           <layer ID="102" name="FE">
            <labels>
             <label ID="10201" end="46" name="Name" start="43"/>
            </labels>
           </layer>
          </layers>
         </annotationSet>
         <annotationSet ID="2" frameName="Event_instance">
          <layers>
           <layer ID="201" name="Target">
            <labels>
             <label ID="20101" end="15" name="Target" start="12"/>
            </labels>
           </layer>
           <layer ID="202" name="FE">
            <labels>
             <label ID="20201" end="15" name="Instance" start="12"/>
            </labels>
           </layer>
          </layers>
         </annotationSet>
         <annotationSet ID="3" frameName="Existence">
          <layers>
           <layer ID="301" name="Target">
            <labels>
             <label ID="30101" end="21" name="Target" start="17"/>
            </labels>
           </layer>
```

```
            <layer ID="302" name="FE">
              <labels>
                <label ID="30201" end="15" name="Entity" start="10"/>
              </labels>
            </layer>
          </layers>
        </annotationSet>
        <annotationSet ID="4" frameName="Time_vector">
          <layers>
            <layer ID="401" name="Target">
              <labels>
                <label ID="40101" end="3" name="Target" start="0"/>
              </labels>
            </layer>
            <layer ID="402" name="FE">
              <labels>
                <label ID="40201" end="15" name="Event" start="10"/>
              </labels>
            </layer>
          </layers>
        </annotationSet>
      </annotationSets>
    </sentence>
```

Not all files could be processed into XML file. For some unknown reasons which I did not have time to debug, SEMAFOR threw exceptions when processing the files. I did not use the files that could not be processed by SEMAFOR.

4.2.2 Calculating similarity between frames

Next, I extracted the frames from an XML file with a Python script. Each folktale was now represented by a list of frames. I treated the frames as words and built a gensim corpora out of the frames. Then I created vectors of frames to represent the story and calculated the folktales' pairwise similarity with cosine similarity measurements. The scores are relatively high, ranging from 0.37 to 0.94.

Some stories that I personally considered similar, such as:

- "Lovely Ilonka" and "The Goose Girl" have a frame score of 0.77
- "Brother and Sister" and "The Six Swans" have a frame score of 0.87
- "Aladdin" and "Jack and the Beanstalk" have a frame score of 0.80
- "Little Red Riding Hood" and "Hansel and Gretel" have a frame score of 0.72

4.3 Similarity between Agents

4.3.1 Identifying the agents

In this project, agents are identified as human participants in the story. (Animal agents do not count in this project.) To extract the agents of the story, I performed two steps: (1) identifying the nouns in the story, and (2) identifying which nouns were persons.

The words in each story were tagged with NLTK part-of-speech (POS) tagger. I selected the words that had tags “NN” (singular noun) and “NNS” (plural nouns) to identify the nouns. The reason I only used these two tags was that it did not make sense to include other noun tags, such as “NNP” (proper noun), “PRP” (personal pronoun), “PRP$” (possessive pronoun). Proper nouns are names, so they are too specific. Pronouns do not give information about the agent’s identity. Overall, I identified 7431 nouns, from “a’-thegither” to “zither-playing”. (Note: The language of the folktales is mostly 19th-century English.) The list of nouns can be seen in the file “nouns.txt”.

Then I proceeded to extract the persons out of these nouns. I used NLTK WordNet library functions to obtain the hypernyms of each word. If a word has “person.n.01” as one of its hypernyms, I identified that word as a person and included it in my list of agents. If a word has several word senses, I iterated over all the senses until it found the hypernym “person.n.01” (if it has any). For example, the word sense “queen.n.01” means the queen bee, so it does not have “person.n.01” as its hypernym. The queen that means “a female monarch” is found in the word sense “queen.n.02”. I discovered 955 agents, from “acquaintance” to “youth.” The list contains false positive words, such as “annoyance”, “arrival”, and “balance”, which, somehow, had “person.n.01” as their hypernyms. For this project, I ignored the false positives.

Out of the 955 agents, I examined for each folktale how many agents they had. In the end, each folktale was represented by a list of agents, in the same way they were represented by lists of frames.

4.3.2 Calculating Similarity between Agents

Initially, I wanted to create vectors of agents and calculate the cosine similarities between the vectors, as I did for the frames. However, the scores yielded by this method were low, ranging from 0.037 to 0.69. After a suggestion from Dr. Katrin Erk, I decided to perform “greedy matching” on two sets of agents. I used NLTK Wu-Palmer similarity measurement because it does not need a corpus as its parameter, and its scores range from 0 to 1.

First, I identified the agents that existed in both sets. They would have similarity scores of 1, so I took them out first. Then, I performed greedy matching on the remaining elements. The first element from the first set was compared to all elements in the second set. After it found its match, i.e. the element from the second set that the highest similarity, the next element from the first set would look for its match from the second set. However, it could not pick an element that had been chosen by the its predecessor.

For example, consider this two sets: [king, princess, nobleman, witch] and [queen, prince, king]. First, I removed “king” from both sets, leaving [princess, nobleman, witch] and [queen, prince]. Next, I allowed the set with less elements to iterate over the set with more elements. Thus, in this example, “queen” from the second set would choose its most similar entry in the first set. It picked “princess” because it had the highest similarity (0.86). The next turn, “prince” would pick its match from the first set. “Prince” was most similar to “princess” (0.90). However, since “queen” had picked “princess”, “prince” could not pick “princess”. It went with the second most similar match, which was “nobleman” (0.86). “Witch” was not picked, and it was paired with a null element with a score of 0.

The total score is the sum of the number of same elements and the sum of the greedy matching results, divided by the length of the longer set. In the example above, the total score is (1 + 0.86 + 0.86 + 0) / 4 = 0.68.

Using this method, the scores now range from 0.005 to 0.85.

- "Lovely Ilonka" and "The Goose Girl" have a score of 0.63 (previously 0.391 without greedy matching)
- "Brother and Sister" and "The Six Swans" have a score of 0.52 (previously 0.56)
- "Aladdin" and "Jack and the Beanstalk" have a score of 0.76 (previously 0.37)
- "Little Red Riding Hood" and "Hansel and Gretel" have a score of 0.30 (previously 0.38)

4.4 Combining the similarity scores from the agents and frames

I calculated the average between the frame scores and agent scores to get the similarity scores.

- "Lovely Ilonka" and "The Goose Girl" have a score of 0.699
- "Brother and Sister" and "The Six Swans" have a score of 0.77
- "Little Red Riding Hood" and "Hansel and Gretel" have a score of 0.51

(It appeared that "Aladdin" got an exception and did not appear in the combined scores list.)

5. Experiments and Results

I ran over the dataset with LDA method to discover the topics of the folktales. I picked 10 topics. I removed the stopwords and words with certain tags such as coordinating conjunctions, preposition, pronouns, and determiners "['CC','DT', 'IN', 'MD', 'TO', 'PRP', 'PRP$', 'NNP', 'NNPS', 'CD']". I added the following words to the stopwords: ['said', "n't", 'then', 'answered', 'thou', 'thee', 'thy', 'so', 'when', 'yes', 'got', 'little', 'asked', 'till'] because they appeared across all topics. However, some words in the topics still overlapped with each other.

Here are the 10 topics and the words:

Topic 0: king(0.012), go(0.009), come(0.008), man(0.008), time(0.006), take(0.006), make(0.006), old(0.005), think(0.005), away(0.005), know(0.005), day(0.005), wife(0.005), get(0.005), back(0.004), princess(0.004), great(0.004), long(0.004), home(0.004), saw(0.004)

Topic 1: go(0.012), make(0.008), prince(0.008), come(0.008), great(0.006), take(0.005), day(0.005), say(0.005), know(0.005), time(0.005), man(0.005), princess(0.005), well(0.004), find(0.004), jack(0.004), give(0.004), sultan(0.004), think(0.004), leave(0.004), much(0.004)

Topic 2: girl(0.008), go(0.008), come(0.008), time(0.007), old(0.006), king(0.006), daughter(0.006), witch(0.006), child(0.006), day(0.006), make(0.005), take(0.005), woman(0.005), well(0.005), know(0.005), head(0.005), leave(0.004), give(0.004), look(0.004), get(0.004)

Topic 3: king(0.015), son(0.008), man(0.007), go(0.006), come(0.006), day(0.006), make(0.005), old(0.005), take(0.005), say(0.005), queen(0.005), time(0.004), young(0.004), let(0.004), leave(0.004), back(0.004), think(0.004), youth(0.004), saw(0.004), away(0.004)

Topic 4: king(0.019), go(0.012), princess(0.011), come(0.009), man(0.008), day(0.007), time(0.007), away(0.005), make(0.005), prince(0.005), old(0.005), take(0.005), horse(0.005), back(0.005), daughter(0.005), long(0.005), home(0.005), find(0.005), great(0.004), know(0.004)

Topic 5: man(0.012), go(0.010), day(0.008), old(0.007), wife(0.007), come(0.007), king(0.007), time(0.006), find(0.006), make(0.006), woman(0.005), back(0.005), take(0.005), know(0.005), leave(0.005), think(0.005), girl(0.004), never(0.004), house(0.004), away(0.004)

Topic 6: go(0.009), come(0.007), king(0.006), back(0.006), man(0.005), know(0.005), day(0.005), time(0.005), head(0.005), tree(0.005), old(0.004), giant(0.004), take(0.004), look(0.004), never(0.004), horse(0.004), wolf(0.004), make(0.004), son(0.004), find(0.004)

Topic 7: go(0.006), gerda(0.006), come(0.006), man(0.005), know(0.005), day(0.005), make(0.005), great(0.005), find(0.004), hand(0.004), cry(0.004), think(0.004), eye(0.004), neangir(0.004), bassa(0.004), look(0.004), old(0.004), soldier(0.004), nothing(0.004), back(0.004)

Topic 8: prince(0.019), king(0.015), princess(0.014), queen(0.008), make(0.007), find(0.007), time(0.006), fairy(0.006), day(0.006), go(0.006), take(0.005), know(0.005), think(0.005), come(0.005), great(0.005), palace(0.004), saw(0.004), back(0.004), look(0.004), much(0.004)

Topic 9: jackal(0.013), come(0.009), go(0.009), make(0.006), take(0.005), lasse(0.005), back(0.005), man(0.005), pivi(0.005), moti(0.005), owen(0.005), know(0.004), horse(0.004), mouse(0.004), day(0.004), castle(0.004), find(0.004), let(0.004), good(0.004), time(0.004)

There are also some names in the topics, such as “gerda” and “owen”. I suspect this is caused by the uneven length of the folktales. Longer folktales dominate the words. A long story like “The Snow Queen” that has a character “Gerda” may mention “Gerda” so often that it appears in one of the topics.

Also, because there are too many common words, it is difficult to distinguish between the topics. I extracted top three most similar stories according to LDA. I present an overview of 3 pairs of stories which are considered similar by LDA and compare them with my method.

1. “The Brownie of the Lake”, considered similar to “Hansel and Grettel”. According to my method, the score is 0.78.
2. “The Slaying of the Tanuki”, considered similar to “Adventures of an Indian Brave” (0.99). According to my method, the score is 0.78.
3. “The Owl and the Eagle”, considered similar to “Mother Holle” (0.99). According to my method, the score is 0.78.

References

[1] Folgert Karsdorp and Antal van den Bosch. 2013. “Identifying motifs in folktales using topic models.” In *Proceedings of the 22 Annual Belgian-Dutch Conference on Machine Learning*, BENELEARN-2013, Nijmegen, The Netherlands, pp. 41-49.

[2] Folgert Karsdorp, Peter van Kranenburg, Theo Meder, and Antal van den Bosch. 2012. “Casting a Spell: Identification and Ranking of Actors in Folktales.” In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pp. 39-50. Lisbon, Portugal: Edições Colibri.

[3] Folgert Karsdorp, Peter van Kranenburg, Theo Meder, Dolf Trieschnigg, Antal van den Bosch. 2012. "In search of an appropriate abstraction level for motif annotations." In *Proceedings of the 2012 Computational Models of Narrative Workshop*, Istanbul, Turkey, pp. 22-26.

[4] Hans-Jorg Uther. 2004. "The Types of International Folktales: A Classification and Bibliography." Academia Scientiarum Fennica, Helsinki, Finland.

[5] Neil McIntyre and Mirella Lapata. 2010. "Plot induction and evolutionary search for story generation." In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1562–1572, Uppsala, Sweden. Association for Computational Linguistics.

[6] Paula Cristina Vaz Lobo and David Martins de Matos. 2010. "Fairy tale corpus organization using latent semantic mapping and an item-to-item top-n recommendation algorithm." In *Proceedings of the 2010 Language Resources and Evaluation Conference (LREC 2010)*, Valletta, Malta.

[7] Takenori Wama and Ryohei Nakatsu. 2008. "Analysis and Generation of Japanese Folktales Based on Vladimir Propp's Methodology." In ECS 2008: 129-137.

[8] Victoria Anugrah Lestari and Ruli Manurung. 2015. "Measuring the Structural and Conceptual Similarity of Folktales using Plot Graphs." In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 25–33, Beijing, China, July 30, 2015.

[9] Vladimir Propp. 1928. Morphology of the Folk Tale. Translated 1968.