Classification of folktales

Dominik Macháček, Martin Banzer

July 17, 2017

1 Introduction

In this paper we contribute to folkloristic research by designing an automatic classifier of texts according to Aaarne-Thompson-Uther (ATU) classification system, which was developed as a standardized way for organization of texts for folkloristics research. With a support of our automatic classifier, researchers can easier study real-world stories transmission and origin. In our work we used the annotated corpus of texts from the Multilingual Folk Tale Database [Jan]. Our work also includes a crawler for downloading this corpus from the Internet.

2 Corpus description

2.1 Aarne-Thompson-Uther index

In the Aarne-Thompson-Uther index (ATU), folktales are divided into 7 groups: animal, magic, realistic, stupid ogre, anecdotes and jokes, formula. These labels are in so called ATU-level-1.

These groups are then subdivided into another groups, as figure 1 shows.

2.2 Multilingual Folk Tale Database

The Multilingual Folk Tale Database (MFTD) [Jan] is a collection of folktales that provides for each tale information about its title, language and ATU group. Since there is no reliable folktale classifier yet, every tale has to be assigned to one of the ATU groups manually by experts. MFTD is open database, everyone can create an account, upload stories to MFTD and label them, even if he or she is actually not an expert. Therefore some of the labels might be disputable. In all groups there is an *other* section, which contains the most questionable tales. Also there are some tales which have no label at all. We collected those tales in a group called *Unknown*.

2.3 Crawler

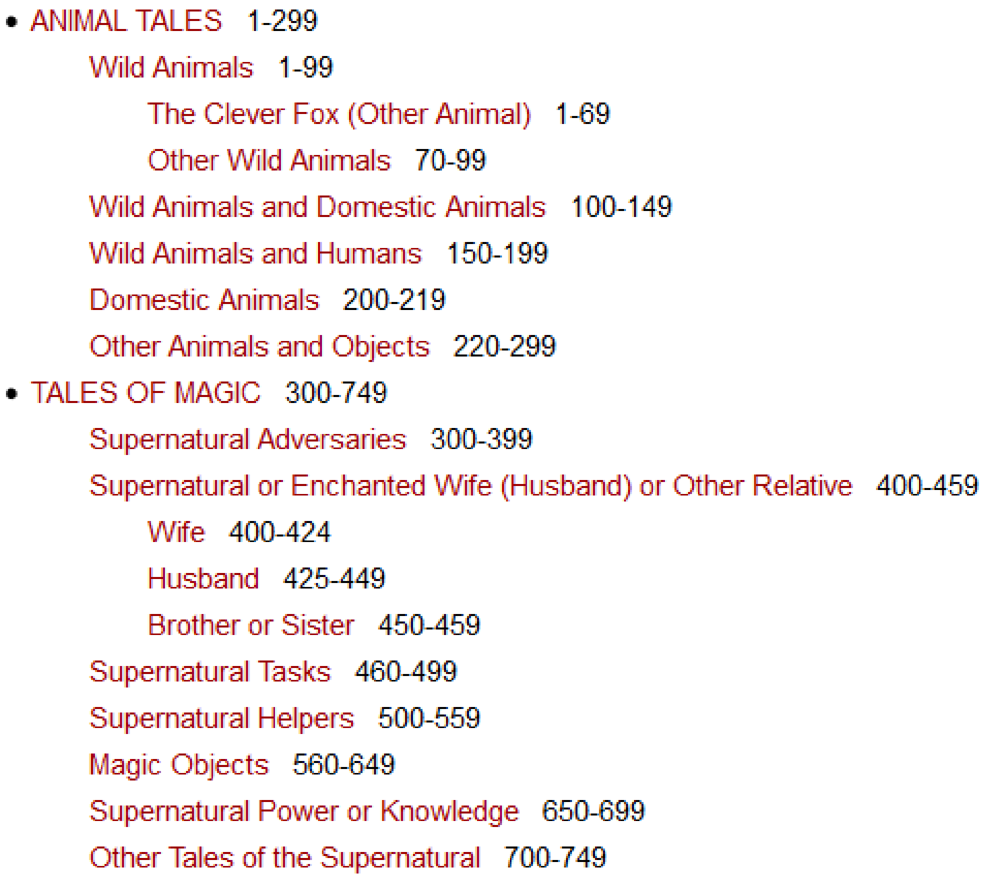
Our colleague Simon Ahrendt provided us with a crawler that covered the Multilingual Folktale Database and extracted folktales with available metadata. This gave us an access to information about the language, the ATU index number and title.

Special acknowledgments to Simon.

2.4 Distribution of languages and labels

The Multilingual Folk Tale Database contains stories in 11 languages. We crawled all of them. Figure 2 summarizes the number of stories by language. In total we have 901 stories in English and between roughly 500 and 600 stories in other 4 languages, French, Spanish, Hungarian and Russian. We realized we have too few stories in German, Danish, Polish, Italian, Czech and Dutch to make a strong supervised classifier for them.

We also realized that most of stories in our Multilingual Folk Tale Database don't have ATU labels assigned. Figure 3 shows the numbers of stories with known ATU level 1 by languages. The language with most labeled stories is English, we have 342 labeled stories. The second is German with 227 stories. This is too few for supervised classifier, therefore we decided to focus our work only to English stories. For other languages we have too few resources.



- ANIMAL TALES 1-299
    - Wild Animals 1-99
        - The Clever Fox (Other Animal) 1-69
        - Other Wild Animals 70-99
    - Wild Animals and Domestic Animals 100-149
    - Wild Animals and Humans 150-199
    - Domestic Animals 200-219
    - Other Animals and Objects 220-299
- TALES OF MAGIC 300-749
    - Supernatural Adversaries 300-399
    - Supernatural or Enchanted Wife (Husband) or Other Relative 400-459
        - Wife 400-424
        - Husband 425-449
        - Brother or Sister 450-459
    - Supernatural Tasks 460-499
    - Supernatural Helpers 500-559
    - Magic Objects 560-649
    - Supernatural Power or Knowledge 650-699
    - Other Tales of the Supernatural 700-749

Figure 1: Example of the classification levels from the ATU. *ANIMAL TALES* and *TALES OF MAGIC* are in the first level, *Wild Animals, Wild Animals and Domestic Animals* and *Wild Animals and Humans* are in the second level and *The Clever Fox* and *Other Wild Animals* are in the third level. The numbers behind each line are so called ATU numbers which belong to each category. They are irrelevant for our work.

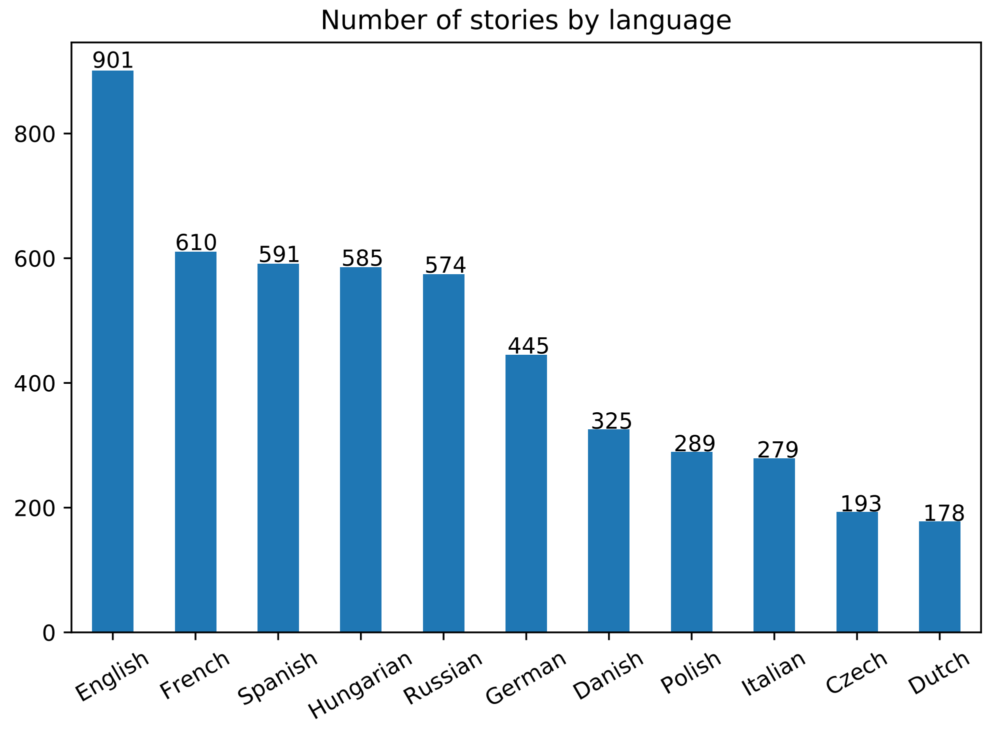


Figure 2: Number of all stories by language.

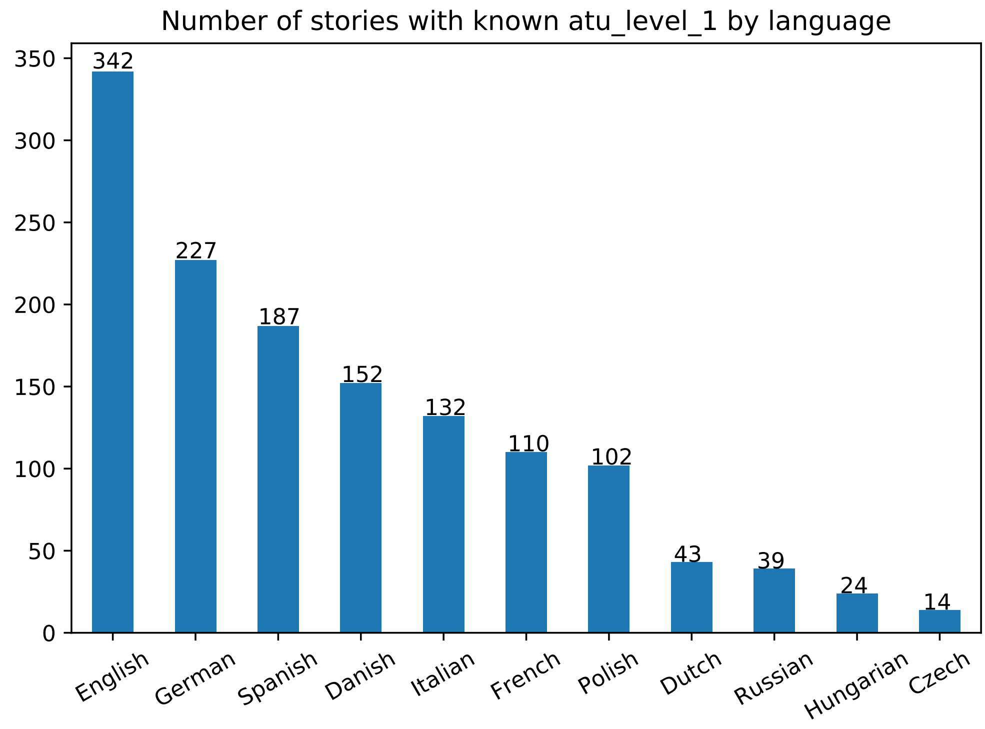


Figure 3: Number of stories with known ATU level 1 by language.

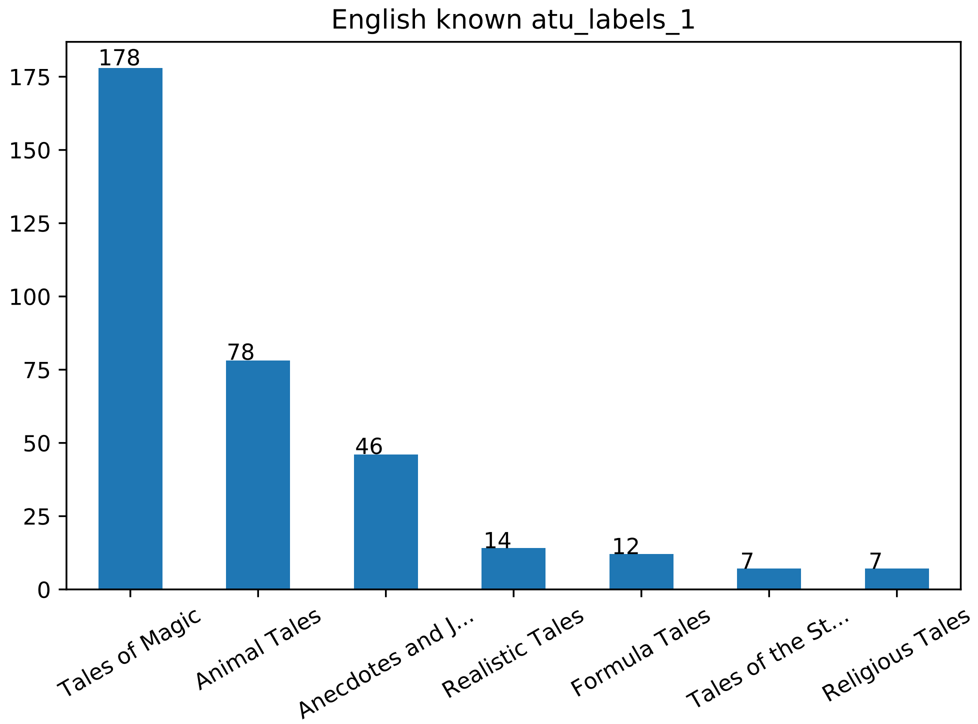


Figure 4: Distribution of English stories by ATU level 1.

Next we focused on the distribution of labels of English stories by level 1 and 2. Figure 4 shows that we have 7 classes by level 1: 176 Tales of Magic, 78 Animal Tales, 46 Anecdotes and Jokes. We have 14, 12, 7 and 7 representatives of other four classes.

Figure 5 describes the distribution of stories by level 2. We realized we have too few representatives of most classes by level 2, therefore we decided to focus only on level 1.

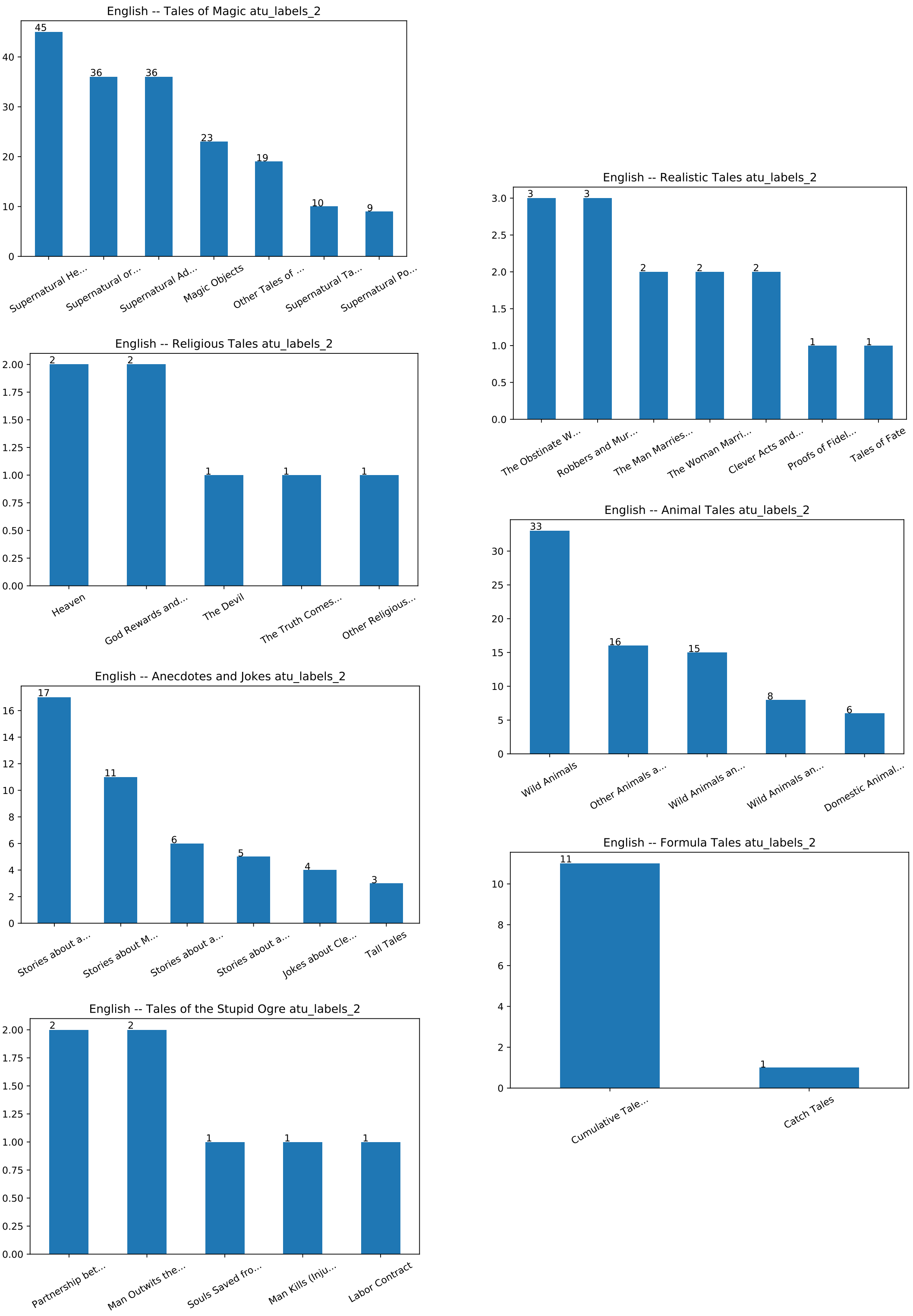
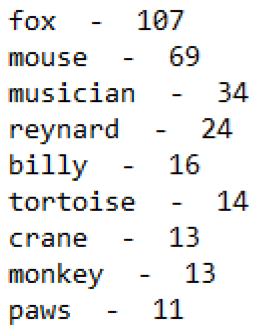


Figure 5: Distribution of English stories by ATU level 2.



```
fox  -  107
mouse  -  69
musician  -  34
reynard  -  24
billy  -  16
tortoise  -  14
crane  -  13
monkey  -  13
paws  -  11
```

Figure 6: The most common animal tale keywords from the texts.

3 Data preprocessing

We split our labeled data to training and evaluation part in ratio 70:30. We made sure that the distribution of classes is the same in both parts.

Then we tokenized all texts with usage of tokenization function from NLTK [NLT] and we removed punctuation and stopwords.

4 Feature design

We designed, implemented and compared several methods for classification.

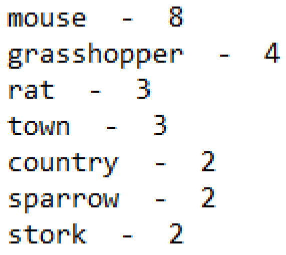
4.1 Trivial keyword extraction

We created two different keywordlists: one for the tale titles and the other for the actual texts.

As an example for our keywordlist creation, we will have a look at animal tales and focus on the tales texts. For this example we computed one list, which counts the appearances of every word from every animal tale. The other list counts the appearances of every word from every tale but the animal tales. Finally we delete every entry from the first list which appears more often in the second list than in the first list.

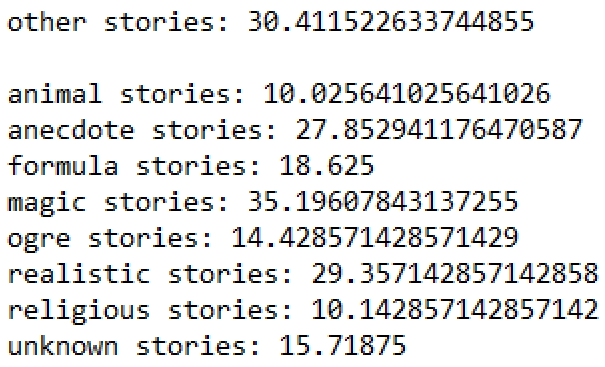
Figure 6 shows the most common keywords from animal tale texts. After we computed keywordlists for every ATU group, we noticed that some lists can be replaced by more intuitive features. For the animal tales example we could also use a list of animal names. We tried this by specifically looking for *animals* using the NLTK WordNet module for Python. But this method also interprets the word *Queen* as an animal because *Queen* appears in the NLTK feature list for animals like *ant queen* or *bee queen.* Therefore we stayed with our keywords.

The keywordlists for the titles proved to be more problematic because the available data is small, due to the fact that every story has only one title and that those titles are very short, the keywordlists for titles are not as effective as the keywordlists for texts. Figure 7 shows the most common keywords from animal tale titles.



```
mouse  -  8
grasshopper  -  4
rat  -  3
town  -  3
country  -  2
sparrow  -  2
stork  -  2
```

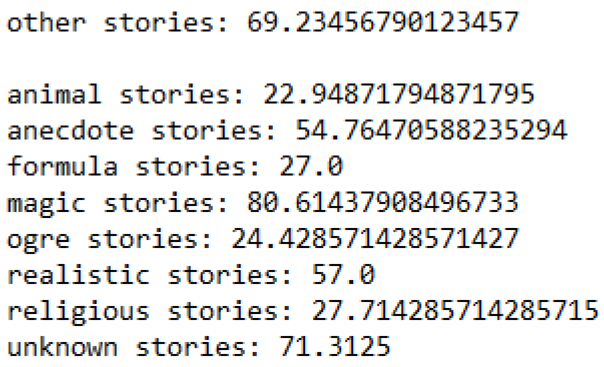
Figure 7: The most common animal tale keywords from the texts.



```
other stories: 30.411522633744855

animal stories: 10.025641025641026
anecdote stories: 27.852941176470587
formula stories: 18.625
magic stories: 35.19607843137255
ogre stories: 14.428571428571429
realistic stories: 29.357142857142858
religious stories: 10.142857142857142
unknown stories: 15.71875
```

Figure 8: The average number of direct speeches per text in the ATU-index groups.



```
other stories: 69.23456790123457

animal stories: 22.94871794871795
anecdote stories: 54.76470588235294
formula stories: 27.0
magic stories: 80.61437908496733
ogre stories: 24.428571428571427
realistic stories: 57.0
religious stories: 27.714285714285715
unknown stories: 71.3125
```

Figure 9: The average amount of sentences per text in the ATU-index groups.

4.2 Information retrieval based keyword selection

Another approach we used was related with tf-idf statistics which is used to reflect the importance of a term in a document with respect to another documents. However, we can't use it because our goal is different. We intend to find terms appearing in all (or at least in much as possible) tales in a class and not appearing in other classes. Therefore we implemented following statistics:

Let C be a total number of occurrences of a term in one class and D the the number of documents in a class containing the term. The statistics we used is then C^D. We select the terms with the highest C^D. This emphasize terms appearing in many tales in a class, but still gets a chance to frequent terms to be ranked higher.

For every class we selected top 100 words by this statistics and excluded singletons. Then we made union of these words.

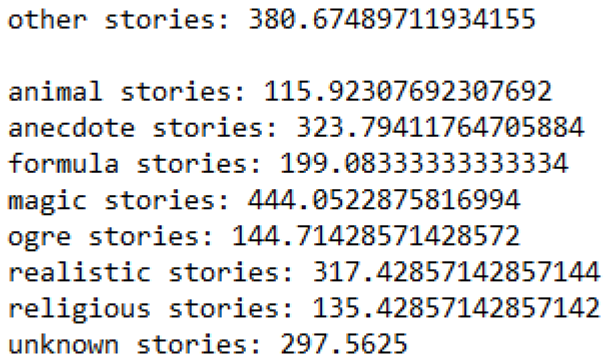
This method selects all important keywords but it also words appearing in all categories which are useless for classification. Therefore we later used greedy forward selection to find the best subset of these keywords.

4.3 Other features

Besides the keywords we also used a list of features which simply count multiple things in every ATU group. We counted for each ATU group the amount of sentences and words, the number of proper names, the number of direct speeches and the number of special signs for texts or for titles.

In the following examples we included a special group called *other stories*. These stories are non-animal stories. Thereby we can compare animal tales to *other stories* which tells us for example, that on average, the length of animal tales is about a quarter of the length of other tales.

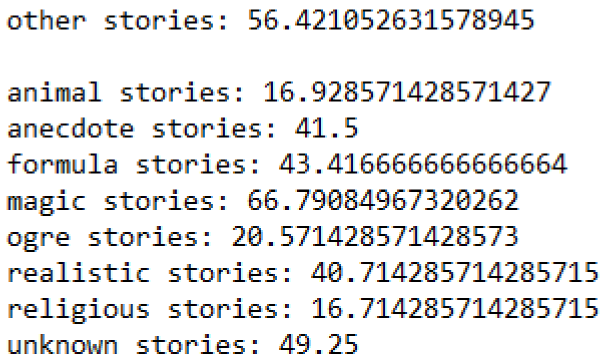
Folgert Karsdorp [Kar16] discusses this topic as well. There are similar results mentioned as we got from our feature inclusion.



```
other stories: 380.67489711934155

animal stories: 115.92307692307692
anecdote stories: 323.79411764705884
formula stories: 199.08333333333334
magic stories: 444.0522875816994
ogre stories: 144.71428571428572
realistic stories: 317.42857142857144
religious stories: 135.42857142857142
unknown stories: 297.5625
```

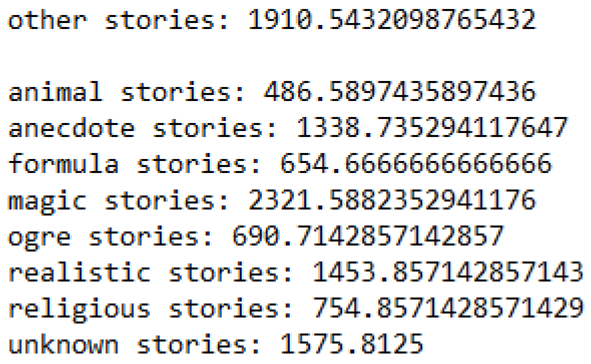
Figure 10: The average number special signs per text in the ATU-index groups.



```
other stories: 56.421052631578945

animal stories: 16.928571428571427
anecdote stories: 41.5
formula stories: 43.416666666666664
magic stories: 66.79084967320262
ogre stories: 20.571428571428573
realistic stories: 40.714285714285715
religious stories: 16.714285714285715
unknown stories: 49.25
```

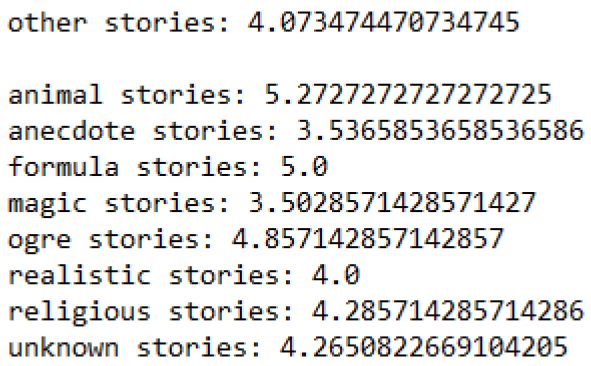
Figure 11: The average number proper names per text in the ATU-index groups.



```
other stories: 1910.5432098765432

animal stories: 486.5897435897436
anecdote stories: 1338.735294117647
formula stories: 654.6666666666666
magic stories: 2321.5882352941176
ogre stories: 690.7142857142857
realistic stories: 1453.857142857143
religious stories: 754.8571428571429
unknown stories: 1575.8125
```

Figure 12: The average amount of words per text in the ATU-index groups.



```
other stories: 4.073474470734745

animal stories: 5.2727272727272725
anecdote stories: 3.5365853658536586
formula stories: 5.0
magic stories: 3.5028571428571427
ogre stories: 4.857142857142857
realistic stories: 4.0
religious stories: 4.285714285714286
unknown stories: 4.2650822669104205
```

Figure 13: The average amount of words per title in the ATU-index groups.

4.4 Summary of feature selection

Here we summarize all feature selection methods we implemented and compared. Numbers with hashtag are used as a reference to table with evaluation results.

```
#1.0 simple most frequent words
-- take 20 mfw for each category
-- values are numbers of occurences in a text
-- not scaled

#1.1
-- same as #1, but
-- MinMaxScaling were used
-- not total number of occurences in a text, but number of occ/text
length

#2.0
-- take 100 words by highest value of C^D
   -- C is a total number of occurences in all text in a class
   -- D is a number of documents in a class containing given word
   -- ^ is power
-- don’t remove words chosen from 2 classes
-- in total, 242 words
-- scaled

#2.1
-- order features by forest features importance
-- use greedy forward selection to find optimal number of features -- it’s 14

#2.2
-- use binary features, not real
    -- 0 or 1: keywords appears at least once in a text
    -- new forward selection
    -- optimal number is 16
```

5 Classification

In previous section we elaborated several methods for design of features. Here we describe the actual classification and evaluation.

5.1 Feature values

Once we have a list of typical words for every category, we have to create a feature matrix. Its rows are the documents and columns are the keywords. We implemented and compared several methods how to fill the matrix values. The values can be the absolute numbers of occurrences of given word in a document. One high value can unbalance the whole classifier, so second option is to scale them to [0,1]-range using min-max scaling. And the third option is to use only binary values zero or one: one if the word appears in the document at least once, zero otherwise.

With these matrices we trained several classifiers from `sklearn` library and evaluated their performance on test set.

Our comparison shows that the binary values give the best performance.

5.2 Classification evaluation results

We used several classification methods from `scikit` library: K-NN, SVM, Decision Tree, Random Forest, AdaBoost and Naive Bayes. We used them with different parameters to find the best settings.

Table 1 summarizes the best results from 7-class classification, it is a problem where one correct label from all 7 classes has to be assigned. It shows that feature set #2.2 is the best, the accuracy with Linear SVM classifier is 78%. Baseline is 51%, it is the accuracy which can be achieved by predicting the most frequent class in all cases.

feature set	**classifier**	**accuracy**
	baseline	0.51
#1.0	SVM	0.71
#1.1	Decision Tree	0.6777
#2.1	Random Forest	0.7333
#2.2	Linear SVM	**0.7778**

Table 1: Evaluation results of 7-class classification.

Furthermore, we did also binary classification for 3 most frequent classes because for other classes we have too few resources. The results are in table 2, 3 and 4.

feature set	**classifier**	**accuracy**
	baseline	0.7444
#2.2	AdaBoost	**0.9333**

Table 2: Evaluation results of binary classification, Animal Tales vs. others.

feature set	**classifier**	**accuracy**
	baseline	0.8667
#2.2	AdaBoost	**0.9111**

Table 3: Evaluation results of binary classification, Anecdotes and Jokes vs. others.

6 Annotation validation

Since our corpus was annotated by anonym, we decided to implement a test for validity of annotation, attempt to find incorrectly or disputably labeled stories and suggest experts to revise them.

We tried two approaches, clustering and leave-one-out cross-validation with cosine similarity.

For both tests we stemmed the texts and removed punctuation and stopwords. Then we created a vector of absolute frequencies of remaining terms for every document.

feature set	classifier	accuracy
	baseline	0.5111
#2.2	AdaBoost	**0.8889**

Table 4: Evaluation results of binary classification, Tales of Magic vs. others.

6.1 Clustering for annotation validation

We used a linkage clustering algorithm with cosine similarity as a similarity measure. We applied it to test set and plotted its result as a dendrogram, see figure 14. The algorithm found 2 big clusters. The yellow cluster consists mostly of Magic tales, but some Magic tales appear in other clusters, and some tales, for example animal tale *The Way Of The World*, appear to be more similar to Magic tales than to other animal tales. Maybe it's incorrectly annotated, maybe it's only an coincidence. We suggest experts to revise all such stories.

6.2 Leave-one-out cosine similarity

With the second test we implemented we attempt to judge a coherence of the actual labeling to classes in training data. For every class and every story in a class, we counted the average cosine similarity to all other stories in the class. Then we printed the result as found outliers. As an outlier we used the inter-quartile range method, outliers are the elements appearing outside the interval $(Q_1 - 1.5|Q_1 - Q_3|, Q_3 + 1.5|Q_1 - Q_3|)$, where Q_1 is the value of first quartile and Q_3 the third quartile.

According to this test, all stories but 3 appear to be correctly assigned. The outliers are summarized in figure 15.

Futhermore, we made a 7-class classifier returning a class with the lowest average cosine similarity to training stories. We evaluated it on test set. Its accuracy was 46.67 %, which is below a baseline 51 %. This method is not optimal for classification.

References

[Jan] Maarten Jansen. Multilingual Folk Tale Database. http://www.mftd.org/index.php?action=atu.

[Kar16] Folgert Karsdorp. *Retelling Stories A Computational Evolutionary Perspective*. 2016.

[NLT] NLTK. Natural Language Toolkit. http://www.nltk.org/.

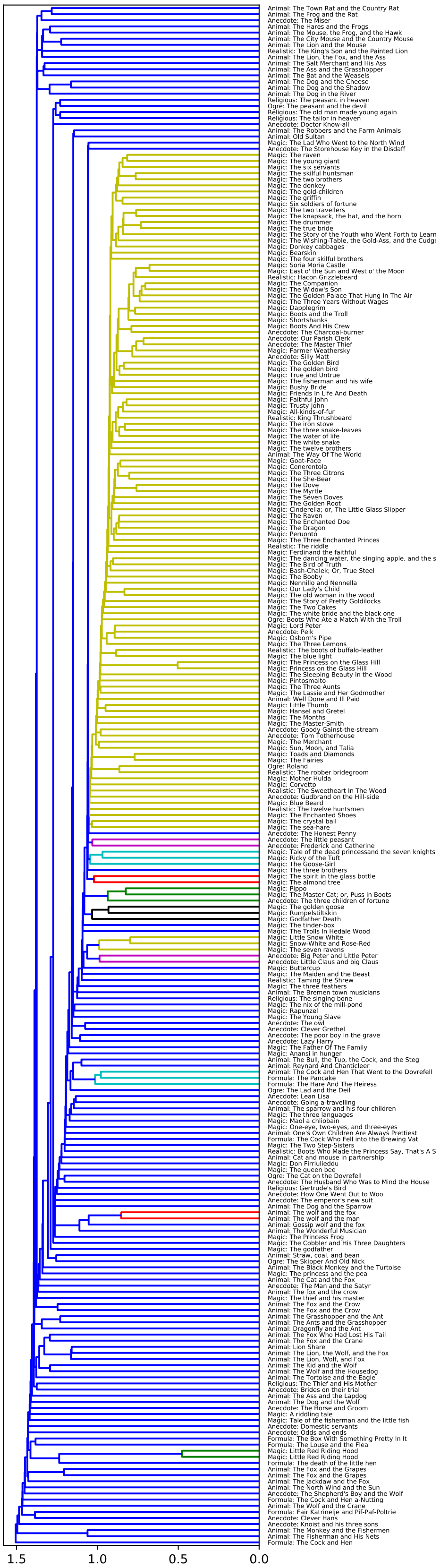


Figure 14: Dendrogram from linkage clustering algorithm. Branch colors represent automaticaly created clusters from clustering algorithm. In branch labels are actual labels and story titles.

```
Format description:
average_cosine_similarity    story_id      story_title



Realistic Tales:
0.2548 5518 The King’s Son and the Painted Lion
--------- IQR lower bound: 0.3020 --------
0.3801 3249 Boots Who Made the Princess Say, That’s A Story
==== First quartile ====
0.3824 3261 Taming the Shrew
0.3886 3305 The Sweetheart In The Wood
0.4123 1885 The boots of buffalo-leather
==== Median ===
0.4164 383 The robber bridegroom
0.4218 640 The twelve huntsmen
==== Third quartile ====
0.4360 212 The riddle
0.4495 498 King Thrushbeard
0.4546 3248 Hacon Grizzlebeard
--------- IQR upper bound: 0.5164 --------

Tales of Magic:
0.1629 1529 A riddling tale
0.2143 4020 Tale of the fisherman and the little fish
--------- IQR lower bound: 0.2246 --------
0.2281 249 Little Red Riding Hood
0.2301 3355 Little Red Riding Hood
0.2388 2015 The princess and the pea
0.2402 649 The thief and his master
0.2517 402 The godfather
0.2556 8096 Maol a chliobain
0.2771 1237 One-eye, two-eyes, and three-eyes
...
```

Figure 15: All possibly incorrectly annotated tales found by leave-one-out cosine similarity test.