Folktale Classification using Learning to Rank

Dong Nguyen, Dolf Trieschnigg, and Mariët Theune

Human Media Interaction, University of Twente, Enschede, The Netherlands
{d.nguyen,d.trieschnigg,m.theune}@utwente.nl}

Abstract. We present a learning to rank approach to classify folktales, such as fairy tales and urban legends, according to their *story type*, a concept that is widely used by folktale researchers to organize and classify folktales. A *story type* represents a collection of similar stories often with recurring plot and themes. Our work is guided by two frequently used story type classification schemes. Contrary to most information retrieval problems, the text similarity in this problem goes beyond topical similarity. We experiment with approaches inspired by distributed information retrieval and features that compare subject-verb-object triplets. Our system was found to be highly effective compared with a baseline system.

1 Introduction

Red Riding Hood, *Cinderella* or the urban legend about the *Vanishing Hitchhiker* are folktales that most of us are familiar with. However, when asking people to recall a specific story, everyone tells his or her own version. Variations of such stories appear due to their oral transmission over time. For example, locations can change, characters can be added, or complete events introduced or left out. In this paper we present work on determining similarity between stories. Our work is guided by the *type indexes* that folktale narrative researchers have developed to classify and to organize stories according to *story types.* A *story type* is a collection of similar stories often with recurring plot, motifs or themes [24].

Many type-indexes have been proposed (see discussions by Uther [27,26]), some tailored to certain narrative genres or geographical locations. In our experiments, we limit our focus to two internationally recognized story type indexes. The first is the frequently used Aarne-Thompson-Uther (ATU) type-index [25] that covers many fairy tales, but also legends, jokes and other folktale genres. An example story type in the ATU index is *Red Riding Hood* (ATU 333). The second is the Type-Index of Urban Legends proposed by Brunvand [6].

The goal, then, of our work is to automatically determine the story types of stories. In particular, we cast this as a ranking problem, where the goal is to assign the highest rank to the most applicable story types. This serves multiple purposes. First, with the increasing digitization of folktales [15,18,1], there is a need to (semi) automate the identification of story types. Second, such a system could help discover new relationships between stories. And as discussed later in

	The Vanishing Hitchhiker (BRUN 01000)
Summary	A ghostly or heavenly hitchhiker that vanishes from a vehicle, sometimes after giving warning or prophecy.
Story 1	A car driver picks up a hitchhiker. They talk about spiritual topics in life. Suddenly the hitchhiker vanishes. The driver tells the story to the police. They tell him that they have heard the story earlier that day as well.
Story 2	A guy bikes through the park at night. He encounters a girl covered in blood. He brings her to the police, but during the trip she suddenly disappears. She resembles a murdered girl.
Story 3	A car driver picks up a hitchhiker and borrows her his sweater. When he stops by to pick up the sweater, he discovers she passed away due to a car accident a while ago. He finds his sweater on her grave.
Story 4	A car driver picks up a girl wearing a white dress. He accidently spills red wine on her dress. He brings her home, and the next day he finds out she died a year ago. When the police open her grave, they find the white dress with the red wine spot.

Table 1. '*The Vanishing Hitchhiker*' and summaries of stories belonging to this type.

this paper, this problem is highly related to problems such as detection of text reuse, plagiarism and paraphrase detection.

To illustrate the concept of a story type, we use *the Vanishing Hitchhiker*, a well known urban legend found in the Brunvand index. Looking at the stories in our dataset[1] classified under this story type, we can identify many variations. A description and example stories are presented in Table 1. The characters (the hitchhiker as well as the driver), can be male or female. Sometimes the clothing of the hitchhiker is described in specific details (e.g. wearing white clothes or a red coat). The particular vehicle also varies (e.g. car, motor, bicycle, horse and carriage). The location can be unspecified, or set in a specific place (city or park). The person that disappears is sometimes described as someone who was murdered, or as an angel. In some variants the clothes of the hitchhiker are found on a grave. Thus, many story elements can be varied.

The goal of this work is to be able to determine the correct story type of a given folktale. This work adds a novel viewpoint on text similarity. Text similarity can be defined on many levels on the similarity spectrum [19], with document identity on the one end, and topical similarity on the other end. Text reuse [19,5], that includes addition, rewriting or removal of text, is viewed as lying in the middle of the spectrum. Story similarity, as we view it, bears many similarities to text reuse. Stories with the same story type can be seen to have originated from a common template/model. However, the similarity goes beyond lexical or topical similarity, in the sense that it is based on events, motifs (narrative elements) and participants of the narrative. Stories are regarded as being of the same type if they match on a more abstract level than just the lexical words (for example locations do not have to match literally), in contrast to text reuse.

[1] The Dutch Folktale Database, a large collection of Dutch folktales, see www.verhalenbank.nl (in Dutch).

We approach this problem using learning to rank and explore features inspired by approaches from distributed information retrieval and features that compare subject-verb-object triplets. We start with a discussion of related work. Then the dataset is presented. Next, we describe the experimental setup and discuss the results. We conclude with a summary and future work.

2 Related Work

Fisseni and Löwe [11] investigated how people perceive story similarity using a user study. Story variations were created by applying character substitutions and varying the order (e.g. reversed temporal order) and style. They found that people focus mostly on motifs, linguistic features and content, and less on the structure of a story when deciding whether two stories are the same.

Friedland and Allen [12] studied the identification of similar jokes. They framed it as a ranking problem like we do here. Their approach used a bag of words model and abstraction of words using manually constructed word lists. In our work we aim to develop an approach that does not rely on manually constructed lists. Their story types were identified heuristically and not motivated by an existing type classification system, while we use existing classification systems used by folktale researchers and consider multiple genres.

As mentioned in the introduction, our problem is similar, but not identical to problems such as identification of text reuse, paraphrasing and plagiarism detection. Clough et al. [10] defined multiple levels of text reuse (wholly derived, partially derived and non derived) and experimented with n-gram overlap, greedy string tiling and sentence alignment using a news corpus. Metzler et al. [19] looked at text reuse on the sentence and document level.

Our problem is also related to the TDT story link detection task [2], that involved determining whether two stories are discussing the same event (e.g. the *Oklahoma City bombing* topic) in the news domain. Most approaches to the story link detection task relied on text similarity. For example the cosine similarity and the clarity metric have been found to be very effective [3,16]. In addition, many approaches focused on matching named entities. However, in our problem, stories do not need to match on exact details such as named entities.

Paraphrase detection [4] involves detecting texts that convey the same information. Used methods include textual similarity measures, as well as using the structure, for example by matching dependency trees. Research in this area has mostly focused on phrases and sentences. The texts in our dataset (as described later) are much longer.

Our problem also shares aspects with plagiarism detection (e.g. see [9]), in particular when certain parts of text are paraphrased. For example Nawab et al. [20] experiment with query expansion using sources such as WordNet and a paraphrase lexicon to measure text similarity on a semantic level. However, some aspects of plagiarism are not applicable to our problem. This holds in particular for cues that identify inconsistencies in text (such as style and vocabulary).

3 Story Type Indexes

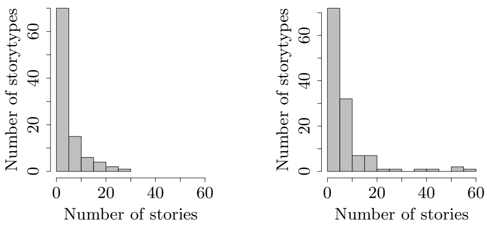
The Dutch Folktale Database is a large collection of Dutch folktales containing a variety of subgenres, including fairy tales, urban legends and jokes. We only consider stories that are written in standard Dutch (the collection also contains many narratives in historical Dutch, Frisian and Dutch dialects). In this paper we restrict our focus to the two type indexes mentioned in the introduction, the ATU index [25] and the Type-Index of Urban Legends [6]. We created two datasets based on these type indexes. For each type index, we only keep the story types that occur at least two times in our dataset. The frequencies of the story types are plotted in Figure 1. Many story types only occur a couple of times in the database, whereas a few story types have many instances.

3.1 Aarne-Thompson-Uther (ATU)

Our first type-index is the Aarne-Thompson-Uther classification (ATU) [25]. Examples of specific story types are *Red Riding Hood (ATU 0333)* and *The Race between Hare and Tortoise (ATU 0275A)*. The index contains story types hierarchically organized into categories (e.g. *Fairy Tales* and *Religious Tales*). We discard stories belonging to the *Anecdotes and Jokes* category (types 1200-1999), since the story types in this category are very different in nature from the rest of the stories[2]. The average number of words per story is 489 words.

3.2 Brunvand

Our second type index is proposed by Brunvand [6] and is a classification of urban legends. Examples of story types are *The Microwaved Pet (BRUN 02000)*, *The Kidney Heist (BRUN 06305)* and *The Vanishing Hitchhiker (BRUN 01000)*. The stories have on average 158 words.



(a) ATU

(b) Brunvand

Fig. 1. Story type frequencies

[2] As was suggested by a folktale researcher. Story types in the *Anecdotes and Jokes* category are mostly based on thematic similarity, while others are based on plot.

4 Experimental Setup

In this section we describe our general experimental setup as well as the specific features used.

4.1 General Setup

Goal We cast the problem of determining the correct story type as a *ranking* problem. Given a story, the system should return a ranking of the story types for that story. We chose a ranking approach, since there are many story types, and most of them only have a few instances in our dataset. In addition, in an actual application new story types could be added over time when new folktales are identified. A ranking of story types is also useful when used in a semi-automatic system: annotators are presented with the list and can choose the correct one. In addition, a ranking of story types can easily be converted into a classification, for example by just taking the top ranked story type as the predicted label.

Evaluation We will evaluate our approach by the *Mean Reciprocal Rank (MRR)*. We use a rank cutoff, by only considering documents in the top 10. We will also evaluate using the *accuracy*, simulating a classification setting. The highest ranked label is then taken as the predicted class.

We use Terrier [21] as our retrieval component, sofia-ml [22] as our learning to rank toolkit and the Frog tool [28] to obtain POS tags (CGN[3] tagset) and dependency tags.

4.2 Baselines

We explore the following baselines, all ranked using BM25:

- *Big document model.* For each story type, we create a big document with the text of all stories of that particular story type. We then issue a query, containing the text of our input document, on these big documents. The result is a ranking of story types. This is similar to the big document models used in Distributed Information Retrieval (e.g. [7,23]), with stories as documents, and story types as collections.
- *Small document model.* For a given story, we issue a query with the text of the story on an index with individual stories. A ranking is returned by ordering story types based on the individual stories that are ranked (duplicates not taken into account). When taking the top ranked label as the class, this is the same as a Nearest Neighbour classifier (k=1).

Experiments showed that the small document approach was more effective than the big document approach (as discussed in the results section). We therefore aim to improve this baseline in our further experiments.

[3] Corpus Gesproken Nederlands (Spoken Dutch Corpus), http://lands.let.kun.nl/cgn/ehome.htm

4.3 Learning to Rank

Compared to traditional information retrieval methods, learning to rank [17] allows researchers to easily add features to their ranking method. We use the sofia-ml toolkit [22] with the SGD-SVM learning algorithm and $\lambda = 0.1$. Using learning to rank, we aim to improve the *small document approach* by incorporating a variety of features. Our proposed method contains the following steps.

- Retrieve an initial set of candidate stories using BM25.
- Apply learning to rank to rerank the top 50 candidates.
- Create a final ranked list of story types, by taking the corresponding labels of the ranked stories and removing duplicates.

4.4 Features

We now describe the features that are used in our learning to rank setting. All features are normalized within a query. We explore features based on lexical similarity, features that match on a more abstract semantic level, and features that reflect the big document baseline.

I Information retrieval measures (IR)
These features indicate the score of the query on the text using the BM25 model. We experiment with three types of queries, resulting in three features: fulltext (*BM25 - Full text*), only nouns (*BM25 - nouns*) and only verbs (*BM25 - verbs*). Note that ranking only on the first feature *BM25 - Full text* results in our small document baseline system.

II Lexical Similarity (LS)
These features represent the similarity of the two texts measured using Jaccard and TFIDF similarity, and calculated on the following token types: unigram, bigrams, character ngrams (2-5), chunks, named entities, time and locations. Location and time words were extracted using Cornetto [29], a lexical semantic database for Dutch, if they were a hyponym of *location* or *timeunit:noun*. The motivation for using these features is that locations (e.g. *house, living room, church*) and time (e.g. *day, September, college year*) can play important roles in the plot of a story.

III Similarity to all stories of the candidate's story type (bigdoc)
This is a feature that resembles the *big document model* as was used in the baseline. This measures the similarity of the input story to the story type of the candidate by taking all stories of that story type into account. As feature we use the retrieval score of the big document of the story type of our candidate story. Again, we experiment with three types of queries, resulting in three features: fulltext (*Bigdoc - BM25 - Full text*), only nouns (*Bigdoc - BM25 - nouns*) and only verbs (*Bigdoc - BM25 - verbs*).

IV Subject Verb Object (SVO) triplets
Events are central to the identity of a story. We aim to capture these using *verb(subject, object)* (SVO) triplets, such as *lives(princess, castle)* or partial triplets such as *disappear(driver,)*. Recently, triplets have been explored to distinguish between stories and non-stories [8]. Triplets are much sparser than just words; we therefore explore allowing partial matches, and abstraction of verbs to a higher semantic level using VerbNet [14].

Triplet extraction For each extracted verb, the system tries to find a matching subject or object by traversing the dependency graph (obtained using the Frog parser) and matching on the relation *su* for the subject, or *obj1*, *obj2* for the object. Only certain POS tags such as nouns, pronouns and named entities are taken into account. Manual inspection showed that the triplets are very noisy, often because of errors by the Frog parser. Each word is replaced by its lemma as given by the Frog parser.

Features To overcome sparsity, we also use features that allow partial matches. For each abstraction level and similarity measure, we create four features representing *Exact* overlap, *Subject-Verb(SV)* overlap, *Object-Verb(OV)* overlap and *Subject-Object(SO)* overlap. We use the Jaccard and TFIDF similarity.

Abstraction Abstraction of triplets reduces the sparsity of the features, and allows stories to match on a more abstract level. We experiment with no abstraction, and abstracting the verbs. Abstraction of verbs is done using VerbNet [14], an English verb lexicon, that groups verbs into 270 general classes. Using relations between Cornetto and Wordnet, a mapping is made between verbs in a story and English verbs. For example, the following Dutch verbs are mapped to the 'consider-29.9' class in VerbNet: *achten (esteem), bevinden (find), inzien (realise), menen (think/believe), veronderstellen (presume), kennen (know), wanen (falsely believe), denken (think).* With verbs, we also experiment with partial matches, but do not add a feature that measures the overlap between subject and object, since these have not been changed.

Reduction of sparsity To illustrate the reduction of sparsity using the methods described, the number of unique elements are shown in Table 2. We find that when allowing partial matches, the number of unique elements decreases a lot (from over 10.000 to 6000-7000). When verbs are abstracted, the counts decrease even more. This is partly caused by verbs that were discarded because VerbNet or Cornetto did not cover them.

Abstraction	Exact	Subject-Object	Subject-Verb	Object-Verb
None (Original)	10260	6325	6416	6925
Verb	8924	NA	4505	5588

Table 2. Number of unique elements - Brunvand (index)

5 Results

5.1 Dataset

For each type index (ATU and Brunvand) we created a dataset. First, we divided the documents into two sets. The query set contains the stories (documents) for which we need to find the story types. The index set contains the stories that need to be ranked. The corresponding labels of these stories can then be used to predict a story type for the query.

Only story types were kept that had at least 2 stories in the folktale database. Then, for each story type one document was assigned to our index set, and one document was assigned to our query set (train/dev/test). The rest of the documents for that particular story type were assigned randomly to either the index or the query set, until the query set was the desired size (e.g. 150 with ATU). The query set then was randomly divided into a train, development and test set while ensuring the desired sizes. Statistics are listed in Tables 3 and 4.

	Index	**Train**	**Dev**	**Test**
Nr. documents	400	75	25	50
Nr. storytypes	98	59	24	43

Table 3. ATU dataset statistics

	Index	**Train**	**Dev**	**Test**
Nr. documents	687	175	50	75
Nr. storytypes	125	92	40	50

Table 4. Brunvand dataset statistics

5.2 Baselines

The results for our baseline methods as described in Section 4.2 can be found in Tables 5 and 6. We find that for both datasets, the smalldoc baseline performs better, although the difference is much larger for the ATU dataset.

	MRR	**Accuracy**
Smalldoc	0.7779	0.72
Bigdoc	0.4423	0.36

Table 5. Baseline results - ATU

	MRR	**Accuracy**
Smalldoc	0.6430	0.56
Bigdoc	0.6411	0.56

Table 6. Baseline results - Brunvand

For our reranking approach, we rerank the top 50 stories obtained using the smalldoc approach. For the ATU, we find that the correct story type is in the top 50 results for 49 out of 50 stories. For the Brunvand index, the correct story type is present in 71 out of 75 stories. This gives an upper bound on the reranking performance and confirms that only reranking the top 50 stories is sufficient for almost all queries.

5.3 Feature Analysis

We evaluate the effectiveness of the feature types by adding them to the baseline model. The results can be found in Tables 7 and 8.

	MRR	Accuracy
Baseline (smalldoc)	0.7779	0.72
+ Bigdoc	0.8367	0.78
+ IR	0.8049	0.76
+ LS	0.7921	0.72
+ Triplets	0.8016	0.72
All	0.8569	0.82

Table 7. Feature analysis - ATU

	MRR	Accuracy
Baseline (smalldoc)	0.6430	0.56
+ Bigdoc	0.7933	0.72
+ IR	0.7247	0.61
+ LS	0.6810	0.60
+ Triplets	0.6600	0.59
All	0.8132	0.76

Table 8. Feature analysis - Brunvand

The performance gains are high compared to the baseline system. The smalldoc baseline had a higher performance on the ATU index, but when including all features the results on the Brunvand index approaches that of ATU.

We also observe that all feature types improve performance. For both datasets the big document features are highly effective. Note that the big document features capture a different type of evidence than the other features. The big document features take similarity to all stories of a particular story type into account, while the other features reflect the similarity between a pair of documents (the input document and the candidate).

Triplets improve performance, but not by much. We analyze the performance of the triplets in more detail by varying the features based on abstraction level and matches as shown in Tables 9 and 10. For both datasets, allowing partial matches when not using any abstraction improves the MRR. However, with ATU the accuracy decreases slightly. Abstraction using verbs does not perform well. When adding both feature types (no abstraction and verb abstraction) the performance does increase.

The performance of the triplets is suboptimal for several reasons. First, manual inspection showed that mistakes of the parser caused triplets to be missed or extracted incorrectly. In addition, we rely on general purpose semantic lexicons such as VerbNet and Cornetto. The coverage of such general lexicons might not be sufficient for specific domains such as folktales.

Abstr.	Matching	MRR	Acc.
No	Exact	0.7762	0.72
No	Exact, partial	0.7902	0.70
Verb	Exact, partial	0.7475	0.68
No, Verb	Exact, partial	0.8016	0.72

Table 9. Triplet analysis - ATU

Abstr.	Matching	MRR	Acc.
No	Exact	0.6422	0.56
No	Exact, partial	0.6556	0.57
Verb	Exact, partial	0.6419	0.56
No, Verb	Exact, partial	0.6600	0.59

Table 10. Triplet analysis - Brunvand

The most important features (i.e. the features with the highest weight) are shown in Tables 11 and 12. We observe that the models learned for ATU and Brunvand have the same features in the top 3. Important features are the big document features and lexical similarity (unigrams, TFIDF). The fact that they share so many features indicates that the ATU and Brunvand indexes are very similar in how story types were defined, and that the same types of evidence are important for finding the correct story types.

Feature	Weight
Bigdoc: BM25 - nouns	0.179
Bigdoc: BM25 - full text	0.158
LS: unigrams - TFIDF	0.109
Bigdoc: BM25 - verbs	0.069
Triplets: SO match, Jaccard, no abstraction	0.063

Table 11. Top 5 important features - ATU

Feature	Weight
Bigdoc: BM25 - full text	0.209
Bigdoc: BM25 - nouns	0.204
LS: unigrams - TFIDF	0.065
IR: BM25 - nouns	0.062
Bigdoc: BM25 - verbs	0.051

Table 12. Top 5 important features - Brunvand

Overall, we believe that the results are very encouraging; a system using all features obtains a high MRR (above 0.8), making this a promising approach to use in a setting where annotators of new stories are presented with a ranked list of possible story types. However, one should keep in mind that we still need to investigate the performance of the approach for other type indexes and texts written in dialects and historical variations.

5.4 Error Analysis

We manually analyzed stories that had a low reciprocal rank using the run with all features.

Both with the Brunvand index and the ATU index, errors occurred because the system found similar stories that matched on the writing style instead of the actual plot. This happened mostly with stories that had a distinguishing style (for example because they were told by the same narrator in a particular setting), and even more when the input story was very short (often with stories of the Brunvand index) or if the correct story type had only a few instances. Thus, if not much content was available to match on plot, our system sometimes incorrectly judged stories to be similar due to style.

With the ATU index, we also observed errors where the system judged stories to be similar because they matched on content words, and not on the actual plot. They might share words related to the location of the story (e.g. *the woods*) or the characters (e.g. *father, son*). This happened in particular with very long stories.

In general, challenging stories were stories with very distinguishing writing styles, and stories with extreme lengths (either very short or very long). Future work should focus on improving performance for these types of stories.

6 Conclusion

This paper presents a study of classifying stories according to their *story types*, a concept used by folktale researchers to organize folktales. Two type indexes were used as the basis of our experiments: the Aarne-Thompson-Uther (ATU) type-index [25] and the Type-Index of Urban Legends [6].

We framed the problem as a ranking problem, where the goal was to rank story types for a given story. We employed a nearest neighbours approach, by ranking individual stories based on their similarity with the given story, and taking the corresponding label as the predicted class. High performance gains were achieved using learning to rank, with features inspired by approaches from distributed information retrieval and features that compare subject-verb-object triplets.

The problem of classifying stories according to their story type presents a new angle on text similarity, and we believe further research on this could also provide new insights into related problems like text reuse, paraphrase detection, story link detection and others. The developed methods could also be useful for classification and organization of other types of narrative data, such as literary fiction, and data reflecting oral transmission, such as interviews [13].

The results were very encouraging, however for such a system to be useful to folktale researchers, stories written in dialect or historical variation should be considered as well. In addition, other story type indexes should also be covered.

Acknowledgements. This research has been carried out within the Folktales as Classifiable Texts (FACT) project, part of the CATCH programme funded by the Netherlands Organisation for Scientific Research (NWO).

References

1. Abello, J., Broadwell, P., Tangherlini, T.R.: Computational folkloristics. Communications of the ACM 55(7), 60–70 (2012)
2. Allan, J.: Topic detection and tracking. Introduction to topic detection and tracking, pp. 1–16. Kluwer Academic Publishers, Norwell, MA, USA (2002)
3. Allan, J., Lavrenko, V., Malin, D., Swan, R.: Detections, bounds, and timelines: Umass and TDT-3. In: Proceedings of Topic Detection and Tracking Workshop (TDT-3) (2000)
4. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. Journal of Artificial Intelligence Research 38(1), 135–187 (2010)
5. Bendersky, M., Croft, W.B.: Finding text reuse on the web. In: WSDM 2009. pp. 262–271 (2009)
6. Brunvand, J.H.: A type index of urban legends. Encyclopedia of Urban Legends. Updated and expanded edition pp. 741–765 (2012)
7. Callan, J.P., Lu, Z., Croft, W.B.: Searching distributed collections with inference networks. In: SIGIR 1995. pp. 21–28 (1995)
8. Ceran, B., Karad, R., Mandvekar, A., Corman, S.R., Davulcu, H.: A semantic triplet based story classifier. In: ASONAM 2012 (2012)

9. Clough, P.: Old and new challenges in automatic plagiarism detection. National Plagiarism Advisory Service (2003)
10. Clough, P., Gaizauskas, R., Piao, S.S.L., Wilks, Y.: METER: MEasuring TExt Reuse. In: ACL 2002. pp. 152–159 (2002)
11. Fisseni, B., Löwe, B.: Which dimensions of narrative are relevant for human judgments of story equivalence? In: The Third Workshop on Computational Models of Narrative (2012)
12. Friedland, L., Allan, J.: Joke retrieval: recognizing the same joke told differently. In: CIKM 2008. pp. 883–892 (2008)
13. de Jong, F.M.G., Oard, D.W., Heeren, W.F.L., Ordelman, R.J.F.: Access to recorded interviews: A research agenda. ACM Journal on Computing and Cultural Heritage (JOCCH) 1(1), 3:1–3:27 (2008)
14. Kipper-Schuler, K.: VerbNet: a broad-coverage, comprehensive verb lexicon. Ph.D. thesis, University of Pennsylvania (2005)
15. La Barre, K.A., Tilley, C.L.: The elusive tale: leveraging the study of information seeking and knowledge organization to improve access to and discovery of folktales. Journal of the American Society for Information Science and Technology 63(4), 687–701 (2012)
16. Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V., Thomas, S.: Relevance models for topic detection and tracking. In: HLT 2002. pp. 115–121 (2002)
17. Liu, T.Y.: Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval, Springer (2011)
18. Meder, T.: From a Dutch Folktale Database towards an International Folktale Database. Fabula 51(1-2), 6–22 (2010)
19. Metzler, D., Bernstein, Y., Croft, W.B., Moffat, A., Zobel, J.: Similarity measures for tracking information flow. In: CIKM 2005. pp. 517–524 (2005)
20. Nawab, R.M.A., Stevenson, M., Clough, P.: Retrieving candidate plagiarised documents using query expansion. In: ECIR 2012. pp. 207–218 (2012)
21. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C.: Terrier: A High Performance and Scalable Information Retrieval Platform. In: SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006) (2006)
22. Sculley, D.: Large scale learning to rank. In: NIPS 2009 Workshop on Advances in Ranking (2009)
23. Si, L., Jin, R., Callan, J., Ogilvie, P.: A language modeling framework for resource selection and results merging. In: CIKM 2002. pp. 391–397 (2002)
24. Thompson, S.: The folktale. Dryden Press (1951)
25. Uther, H.J.: The Types of International Folktales: A Classification and Bibliography Based on the System of Antti Aarne and Stith Thompson. Vols 1-3. Suomalainen Tiedeakatemia, Helsinki (2004)
26. Uther, H.J.: Type- and motif-indices 1980-1995: An inventory. Asian Folklore Studies 55(2) (1996)
27. Uther, H.J.: Classifying tales: Remarks to indexes and systems of ordering. Folks Art - Croatian Journal Of Ethnology and Folklore Research (2009)
28. Van Den Bosch, A., Busser, B., Canisius, S., Daelemans, W.: An efficient memory-based morphosyntactic tagger and parser for Dutch. In: Computational Linguistics in the Netherlands: Selected Papers from the Seventeenth CLIN Meeting, pp. 99–114. OTS (2007)
29. Vossen, P., Hofmann, K., Rijke, M., Tjong, E., Sang, K., Deschacht, K.: The Cornetto database: Architecture and user-scenarios. In: DIR 2007 (2007)