Annotation and Classification of Locations in Folktales

Matthias Lindemann[1], Stefan Grünewald[1] and Thierry Declerck[2]

[1] Department of Language Science and Technology
Saarland University
E-mail: {mlinde|stefang}@coli.uni-saarland.de

[2] DFKI GmbH, Multilingual Technologies
E-mail: declerck@dfki.de

Abstract

In the context of a software project dedicated to the automated classification of folk and fairy tales, we focused on their segmentation by scenes and their respective locations. In contrast to novels, fairy tales are often taking place at the same types of locations, such as castles, in the forest, in a small hut, and the like. That is, locations can be considered as a feature for supporting the general classification of folktales. In this paper, we describe our first annotation approaches for supporting the automatic detection of locations in folktales that are in German language. To our knowledge, this is the first work on automatically detecting locations in folktales.

1 Introduction

In the context of a software project conducted at the Department of Language Science and Technology of the Saarland University we were dealing with the classification of folktales along the lines of schemes proposed by [2], [5] or [6]. One group focused on testing the relevance of segmenting tales by their described scenes. An important aspect of a scene is the location in which it takes place. Contrary to other literary genres, fairy tales seem to have recurrent locations across stories, like castles, forests, small huts, etc. The occurrence of locations can thus be considered as a feature that supports the classification of tales.

We started an investigation on this topic and concentrated in a first step on creating a corpus the annotation of which aiming at supporting the automated detection of locations in folktales written in German language. The task of location detection can be divided into three subtasks, whereas in this paper we only cover the first two subtasks:

1. Recognition of the scene boundaries ("segmentation")
2. Recognition of the type of location where the scene takes place ("classification")
3. Recognizing whether two identical locations from different scenes are the same location ("identity")

For the creation of the corpus we wrote a crawler and downloaded text from 41 collections of tales, with a total of 1880 stories in German from all over the world. The main source is *Projekt Gutenberg*[1]. We excluded very small collections and lyric folktales because they differ much in style. The corpus we assembled from the web crawl contains about 4,3 millions tokens.

For our work on location detection, we first needed to check for which types of locations we could gain enough training data for applying a statistical approach. For this, the corpus has been tagged with the help of the TreeTagger[2] and we looked for the most frequent nouns expressing a location. We were also interested in knowing if a scene is occurring within or outside a location. However, for most types of locations we decided that they are too infrequent and would result in sparsity issues. Therefore, we make this distinction only for the locations "house" and "castle". This corpus is the basis for the different types of annotation we are providing: manual and automated.

2 Manual Annotation

2.1 Annotation Guidelines

We established annotation guidelines for the annotation of segments and locations in tales. The main objective was to find and mark segments in which maximally one location is "involved". But we also allow to mark segments in which more than one location is "involved", in case it is not possible to avoid it.

Following those guidelines three tales have been annotated by six project participants. One tale was taken from the Grimm collection, one tale is by Andersen and one tale was taken from the "One Thousand and One Nights". After this first annotation exercise, we adapted the guidelines in order to respond to encountered issues and problematic cases.

In the new version of the guidelines, a more precise specification for "segment" was given: segment boundaries are given by punctuation signs (excepting commas) and paragraph boundaries. This made it easier to agree on the same level of granularity. We derived 24 different types of locations from the corpus[3]. Table 1 illustrates examples from the guidelines, here translated into English.

---

[1] `http://gutenberg.spiegel.de/genre/marchen-fairy`

[2] cf. [4] and `http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/treetagger.html`.

[3] `Turm, Wüste, Küche, Saal, Schloss_innen, Schloss_aussen, Wald, Haus_aussen,`

Location	Description
Desert	A desert, sand/stone, not in a metaphorical sense
Kitchen	A kitchen (i.e. a room on its own) e.g. in a castle or house
Hall	Hall, ballroom, throne room in a castle,...
Castle_inside	bedroom, private room or study in castle, e.g. a chamber, possibly also a corridor, stairs,....
Castle_outside	castle, palace, villa from outside, i.e. in the open air, balcony or inner courtyard
Church	Church, religious buildings
Nowhere	No place; as if it hadn't been annotated or if having an off-voice, like in a film.

Table 1: Excerpt from the guidelines: which places are to be interpreted in which way.

2.2 Inter-Annotator Agreement

To calculate the inter-annotator agreement, we have chosen Cohen's κ, which is computed pairwise between the annotators[4]. However, it is not directly applicable to cases where each instance (i.e. segment) contains more than one label (type of location). [3] adjust the calculation of the κ statistics so that instances can have a main label and a secondary label with different weights. As this is not the case in our work, we just generalize the calculation for n labels per instance, each with the same weight. However, it is not possible to directly compare two scenes because their corresponding segments will only be the same if the annotators fully agree on the boundaries. To work around the problem, we have used every word occurring in a selected segment as an instance that carries the labels. Table 2 shows the details of the pairwise inter-annotator agreement computation on a small sample of three stories.

	Annot_1	Annot_2	Annot_3	Annot_4	Annot_5
Annot_2	**0.686**				
Annot_3	0.569	**0.718**			
Annot_4	0.519	**0.587**	0.545		
Annot_5	0.469	**0.572**	0.505	0.445	
Annot_6	**0,655**	0.563	0.499	0.405	0.27

Table 2: Pairwise inter-annotator agreement at word level on three folktales after the adaptation of the guidelines

`Haus_innen, Weg, Stadt, Garten, Feld, See, Fluss, Meer, Höhle, Zelt, Stall, Kirche, Gefängnis, Wirtshaus, Mühle, Nirgendwo.` Which translates to *Tower, Desert, Kitchen, Hall, Castle_inside, Castle_outside, Forest, House_outside, House_inside, Way, City, Garden, Field, Lake, River, Sea, Cave, Tent, Stable, Church, Prison, Inn, Mill, Nowhere.*

[4] See [1] for a discussion of Cohen's κ and other methods for measuring agreement among corpus annotators.

3 Towards the Automatic Segmentation

3.1 Features

We represent a segment by a bag of features. To reduce data sparsity compared to using bare words, we tagged and lemmatized all words with the TreeTagger [4] in a preprocessing step, focusing then mostly on open class words. We also use the lemmatized words to record the information whether they have occurred within literal speech (adding to the lemma a quotation mark). This way, we can differentiate between locations being mentioned by the narrator or by a character of the tale. Moreover, prepositions and their corresponding noun phrase's heads seem too important for the identification of locations to lose their connection by the bag of features assumption. Thus, we merged the preposition and its noun phrase's head.

For example, from the passage *"Wie kannst du es wagen," sprach sie mit zornigem Blick, "in meinen Garten zu steigen?"*[5] the following features are extracted: `{können", wagen", sprechen, zornig, Blick, in_Garten", steigen"}` (*{can", dare", say, angry, gaze, in_garden" , climb"}*). In addition, we also mark when a noun is modified by a negative expression like "kein" (*none*), as in "Aber es war kein Meer zu sehen" (*but there was no sea to be seen*). Here the extracted feature is `!_Meer` and not `Meer`.

3.2 Segmentation

The basic idea for the automatic segmentation is that scenes are associated with characters through time and space and that scenes boundaries correspond somehow to changes of temporal and location information. That is, as soon as a movement of the main characters or some time is passing/jumping, a scene boundary must exist. Additionally, we assume there can be no scene boundary within direct speech. Regular expressions are used to determine whether there is any movement, for example `"( |heim|zurück|um|wieder)(kehren|gekehrt)"`[6]

A list of verbs expressing movements was extracted from the corpus also considering frequency information. Imperative forms of such verbs are extracted from the literal speech. Another strategy will have to be implemented for detecting time jumps, as those are typically marked by phrases.

In automatic segmentation by locations, the input text is first separated at punctuation marks or paragraph boundaries on which, according to the guideline, it is possible to segment (see Section 2.1) and, in a second step, it is reassembled anywhere where neither movement nor a time jump are observed. If there indeed is a movement or time jump detected, the task is to decide whether the segment in question (the one with movement or time jump) should be attached to the previous or to the following segment. To illustrate this procedure, the following example has

[5] In English: "How dare you," she said with angry gaze,"to climb into my garden?"

[6] This matches infinitive and participle forms of *return* and *return home*

three provisional segments one of which contains a movement. Since the movement verb **go** is used *at the beginning* of the second segmentation unit, the segmentation algorithm decides to transfer the second block to the third one[7].

> “[...] setz dich darunter und warte, bis die Nacht kommt, so wirst du schon das Gruseln lernen.”
>
> Da **ging** der Junge zu dem Galgen, setzte sich darunter und wartete, bis der Abend kam.
>
> Und weil ihn fror, machte er sich ein Feuer an;

4 Classification

We implemented three approaches for the classification: rule-based, statistical and hybrid. The rule-based classifier applies a keyword-spotting method on the features. To classify a segment as a location, at least one feature of the segment must match a specific regular expression. At the same time, it must not match another regular expression (a kind of blacklist). We call a feature a *key feature* if it fulfills these requirements. This blacklisting is used to cope with German compound nouns. For instance, a simplified rule is[8]:

$$\texttt{(ins?|im)\_.*[Hh]aus} \wedge \neg \left[\texttt{(Gottes|Schnecken|Vogel)haus}\right] \rightarrow \textsc{Haus\_innen}$$

Key features of this rule are for instance "in_Haus", "im_Räuberhaus", but "Haus", "in_Schneckenhaus" etc. are rejected. If several rules apply to the same segment, the rule-based classifier chooses the location with highest prior probability.

The generated corpus is a necessary prerequisite for the use of statistical methods. However, as we do not have labeled training data in necessary quantity, the rule-based classifier must be used to first annotate the corpus. We selected a Naïve Bayesian approach for training the model, and for this crossed the corpus with a window of seven features both the the left and to the right. Whenever a key feature appears in the centre of the window, the content of the window is evaluated as a joint observation of the classified location with the features. We have observed that models are better when they use a context window that distributes weight unevenly so that features further away from the key feature in the middle have a lower weight.

We implemented two versions of this approach, a “simple” one and one with two stages that first performs a binary classification task (BUILDING or NON-BUILDING) to narrow down the set of possible classes which the simple approach has to chose from. Since the training data for this classifier is also generated with a

[7]In English: “”[...] sit underneath it and wait for the night to come, so you will find yourself to learn the fear.” Then the boy **went** to the gallows, sat underneath and waited until the evening came. And because he was freezing cold, he started a fire;

[8]Translations: Gotteshaus - house of prayer, Schneckenhaus - snail shell, Vogelhaus - birdhouse

rule-based system, we can now write rules that can identify buildings but are not specific enough to identify the type of the building, for instance[9]:

`.*([Zz]immer|[Dd]ach|[Ff]enster)` → BUILDING

Instead of keeping the rule-based and statistical approach separate, we also combine them, since the rule-based approach is relatively precise, but in return does not make a statement for some segments. The classification procedure is as follows: First, the rule-based classifier is applied. If there is exactly one result, this location is predicted; if there are multiple results, the statistical classifier with two stages is applied but restricted to the set of locations that the rule-based classifier found. If the rule-based approach does not find any location at all then all location types are taken into account by the statistical model.

5 Evaluation

For the purpose of evaluation, we created a development set consisting of the three folktales (Annot_2, see Section 2.2) and 10 additional locally segmented and annotated tales from the corpus (180 annotated segments). There are two simple methods of evaluation, the first one being the evaluation of the classification with the usual metrics. For that, the segmentation has to be given. The second one is a joint evaluation of segmentation and classification, i.e. calculating agreement. We pursue both methods.

For calculating accuracy we consider a classification to be correct if the predicted label is in the set of the annotated labels.[10] On average, there are 1.144 labels per segment. Here we display in Table 3 a small summary of the evaluation when the segmentation is given.

Approach	**Accuracy**	**Mean F-Score**
Majority class	0.15	0.01
Rule based	0.45	0.41
Naive Bayes	0.43	0.38
Two-stage Naive Bayes	0.43	0.32
Hybrid	**0.53**	**0.46**

Table 3: Accuracy and (arithmetic) mean f-score over all classes of different approaches

The good performance of the hybrid model can be explained: as long as there is only one key feature, the rule-based classifier is applied. In the case of several results, the statistical approach is applied among the hits for taking a better informed decision.

[9]Translations: Zimmer - room, Dach - roof, Fenster - window

[10]This is a simplification, of course. It might be, that the location of a scene cannot be disambiguated but it has to be consistent over the tale.

		Annot_1	Annot_2	Annot_3	Annot_4	Annot_5	Annot_6
Manual	Annot_2	**0.686**					
	Annot_3	0.569	**0.718**				
	Annot_4	0.519	**0.587**	0.545			
	Annot_5	0.469	**0.572**	0.505	0.445		
	Annot_6	**0.655**	0.563	0.499	0.405	0.27	
Automatic	Two_stage_NB_seg	0.327	0.382	0.369	0.362	0.271	0.259
	Two_stage_NB	0.279**	0.295**	0.361**	0.319**	0.057	0.231**
	Hybrid_seg	0.395	**0.55**	**0.526**	**0.418**	**0.312**	0.296
	Hybrid	0.243	0.264	0.26	0.24	-0.008	0.225
	NB_seg	0.309	0.363	0.312	0.343	0.29	0.218
	NB	0.214	0.24	0.226	0.22	0.197**	0.194
	Rule-based_seg	**0.421**	0.5	0.405	0.329	0.272	**0.33**
	Rule-based	0.272	0.276	0.249	0.256	0.152	0.224

Table 4: Agreement between annotators and models. *NB* stands for Naive Bayes and *seg* means that a gold segmentation was given. The best agreement without the gold segmentation marked with **.

Table 4 compares the inter-annotator agreement on the three folktales between manual annotation and automatic annotation. Classifiers with *seg* don't have to call the automatic segmentation but receive the segmentation of Annot_2. The most striking difference in agreement is to be noticed when comparing the same classifier with automatic segmentation and with gold segmentation. Consistent with the good accuracy of the hybrid model on the development set, it performs well in terms of agreement. When comparing agreement of different classifiers to each other, one should be aware that they get the same segmentation (gold or automatic) and their difference in performance is a combination of accuracy against the human annotator and the length of correctly annotated segments, since our way of calculating agreement favors agreement on long segments more than agreement on short segments.

5.1 Error Analysis

There are two major sources of errors that can be identified. Firstly, the automatic segmentation can ignore an actual boundary or detect a boundary where there actually is none. The latter case is especially bad because it results in many small segments that particularly hard (if at all) to classify. Secondly, there can also be errors that originate from the classification.

A large source of error in the automatic segmentation is the coarse way we detect movements, which does not take mood into account and does not disambiguate verbs that can express a movement. For instance, *came to his mind* does of course not entail an actual movement. Similarly, the intention of returning does not necessarily mean a movement.

> Endlich **kam** es ihm in den Sinn, er wollte zu seinem Vater **zurückkehren**.
> *Finally, it **came** to his mind that he wants to **return** to his father.*

Finally, we currently cannot disambiguate whether it is a main character or a minor character that moves (something) to a different location.

Unter Andern **ging** auch einer des Weges dahin, der eine Kuh zu Markte **trieb**.
One of the people who was ***going*** *along the road* ***drove*** *a cow to the market.*

6 Towards a Visualization of the Segmentation by Locations

Related to the investigations described in the preceding sections, some work has been dedicated in setting the bases for a possible automated visualization of the provided annotations. We focused on two aspects:

1. A representation of the scenic structure of a tale
2. a visualization of interactions between characters

6.1 Scenic Structure

The scenic structure of a tale can be represented as a linear graph: The individual scenes form the nodes of the graph, and two nodes are connected by an edge if one scene immediately succeeds the other. Optionally, the graph can also be labeled: Nodes are then annotated with the type of location of the scene as well as the characters involved in it, while edges are annotated with the text of the scene transition. Furthermore, the types of locations can be illustrated with clip art images (e. g. a drawing of a castle for the location type "castle"). Figure 1[11] shows a part of such a graph for the tale Hänsel und Gretel.

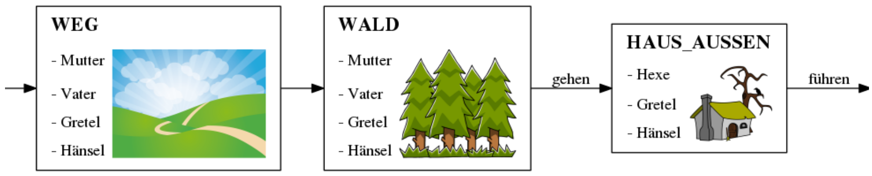

Figure 1: Representing a succession of scenes in Hänsel und Gretel, automatically generated from the annotations. Including types of locations and using clip art images for representing those.

In order to generate a graph as displayed in Figure 1, we are using a Python script that iterates over the annotated scenes of the tale and that for each scene creates a node and its labels, using the DOT graph description language. We use the advanced feature of HTML node syntax to properly arrange the various parts of the node (locations, characters, image). In a final step, code is created which links the nodes with edges to form a linear chain.

6.2 Interactions between Characters

The interactions between the characters in a tale can also be represented in a graph. In this case, every character in the tale is represented by exactly one node, and a edge is drawn from character A to character B if A talks to B at least once over the course of the narrative. The edge is then labeled with the number of times B is addressed by A. Additionally, nodes are

[11] The locations in the nodes are Path (*WEG*), Forest (*WALD*) and House_outside (*HAUS_AUSSEN*).

positioned in such a way as to minimize the distance between characters who interact with each other more frequently. Naturally, unlike the scene graph, such a character interaction graph will in general not be linear. Figure 2 shows an example graph of this kind for the tale “Die Bremer Stadtmusikanten” (*Town Musicians of Bremen*)[12]

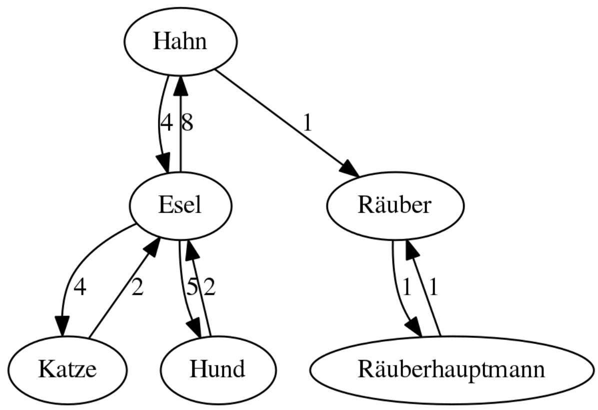

Figure 2: Representation of the interaction between characters in a tale, taking into consideration the frequency of such interactions.

As with the scene graph, we use Graphviz to create the character interaction graph. To extract the necessary information from a tale, we use a nested loop to iterate over all dialogue acts in each scene, ignoring passages spoken by the narrator. For each ordered pair (A,B) of characters, we count how often A talks to B. We then create a node for each character and link it via outgoing edges to all the nodes corresponding to characters they talk to at least once. By adjusting the weight attribute of the edges, we assure that characters that interact frequently are positioned close to each other.

Character interaction will be investigated in more details, as we assume that characters of a folktale interacting with each other are sharing a location, a feature that can improve our current algorithms for their detection.

7 Conclusion

We presented current work in establishing a corpus for supporting the automated classification of locations in folktales. The classification of locations can probably play a relevant role in the classification of tales along the lines of widely used classification systems for narratives. Automatic classification can help with that and provide means of finding spatial patterns in the structure of folktales. We are working on improving the currently implemented classification approaches and extending it to identifying identity of locations. We started also to apply basic algorithms for visualizing tales along their segmentation by locations. We are also aiming at adapting and integrating our annotation scheme with work proposed for example by [7].

[12] The characters in the nodes are a rooster (*Hahn*), a donkey (*Esel*), a cat (*Katze*), a dog (*Hund*), robbers (*Räuber*) and a robber chief (*Räuberhauptmann*).

Acknowledgement

DFKI contribution to this worked has been partially funded by the project QT21, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 645452.

References

[1] Ron Artstein and Massimo Poesio. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596, December 2008.

[2] Vladimir Propp. *Morphology of the folktale*. Trans., Laurence Scott. 2nd ed., University of Texas Press, 1968.

[3] Andrew Rosenberg and Ed Binkowski. Augmenting the kappa statistic to determine interannotator reliability for multiply labeled data points. In *Proceedings of HLT-NAACL 2004: Short Papers*, HLT-NAACL-Short '04, pages 77–80, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[4] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In Daniel Jones and Harold Somers, editors, *New Methods in Language Processing*, Studies in Computational Linguistics, pages 154–164. UCL Press, London, GB, 1997.

[5] Stith Thompson. *Motif-index of folk-literature: A classification of narrative elements in folktales, ballads, myths, fables, medieval romances, exempla, fabliaux, jest-books, and local legends*. Revised and enlarged edition, Indiana University Press, 1955−1958.

[6] Hans-Jörg Uther. *The Types of International Folktales: A Classification and Bibliography. Based on the system of Antti Aarne and Stith Thompson*. Suomalainen Tiedeakatemia, 2004.

[7] Gabriel Viehhauser-Mery and Florian Barth. Towards a digital narratology of space. In *Digital Humanities 2017: Conference Abstracts*, Montréal, Canada, August 2017.