Interpretable Deep Learning Methods for Classifying Folktales According to the Aarne-Thompson-Uther Scheme

Duarte Pompeu, Bruno Martins, and David Matos

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
{duarte.pompeu, bruno.g.martins, david.matos}@tecnico.ulisboa.pt

Abstract. Folktales are works of historic and literary importance, reflecting the culture of those who created and preserved them. There has been limited research regarding the automated analysis of this type of text, using mostly linear models and bag-of-words representations for automatic classification. However, recent research shows that neural network models comparatively achieve a superior performance for similar NLP tasks. As such, we propose to evaluate the use of a cross-language neural network approach based on the previously proposed Hierarchical Attention Network to classify multi-lingual folktales, as well as to explain predictions by generating visualizations. Experiments conducted on the Multilingual Folk Tale Database show an increased performance over baselines when predicting Aarne-Thompson-Uther categories, and we demonstrate the usefulness of neural attention as a method for generating intuitive visualizations of results.

Keywords: Computational Folkloristics, Text Classification, Deep Neural Networks, Cross-Language Learning

1 Introduction

Folktales are stories created by individuals or communities which are used for teaching, disciplining, or entertaining. Appearing in forms such as tales, proverbs, or jokes, they reflect the culture of those who preserved and shared them. As such, they are an important object of study in the fields of Literature and History.

The formal study of this topic is referred to as Folkloristics, which deals with the collection, archival, and analysis of folktales [1]. Enabled by advances in technology and networking, researchers have recently collected and published these works into digital collections such as the Dutch Folktale Database [2], the Archive of Portuguese Legends (APL)[1], and the Multilingual Folk Tale Database (MFTD)[2], which is used in the experiments reported on this paper.

To organize and classify folktales, distinct indexes are commonly employed. One notable example is the Aarne-Thompson-Uther (ATU) classification system,

[1] http://www.lendarium.org
[2] http://www.mftd.org

Table 1: The categories and subcategories from the ATU hierarchy.

Categories	**Subcategories**
Animal tales	Wild animals, Wild and domestic animals, Animals and humans, Domestic animals, Others
Tales of magic	Adversaries, Enchanted relative, Supernatural Tasks, Supernatural Helpers, Magic Objects, Power or Knowledge, Others
Religious tales	God, Truth, Heaven, Devil, Others
Realistic tales	Marrying the princess, Marrying the prince, Fidelity and innocence, The obstinate wife, Good precepts, Clever acts and words, Tales of fate, Robbers and murderers, Others
The stupid ogre	Labor contract, Man and ogre, Man against ogre, Man kills ogre, Man frightens ogre, Man outwits the devil, Souls saved from devil
Anecdotes/Jokes	Fools, Married couples, Women, Men, Religion, Tall tales, Others
Formula tales	Cumulative tales, Catch tales, Others

composed by 2400 different index nodes, and organized in a complex hierarchy of genres, sub-genres, and so on. The first two levels of the hierarchy, which will be referred to as categories and sub-categories, are displayed in Table 1.

In connection to Computational Folkloristics, there has been research regarding automatic classification of folktales. For instance, machine learning models have been used to classify works from the Dutch Folktale Database, namely SVMs [3], as well as Learning to Rank methods [4], showing moderate success in this task. However, in other text classification tasks, this type of algorithms (i.e. linear models leveraging bag-of-words features) is often outperformed by other machine learning models. One such example are deep neural networks, which have been used for natural language processing (NLP) tasks such as classification and sentiment analysis [5]. There has also been research in providing insights to these models' predictions, e.g. by highlighting words which internally produced higher attention weights for a given classification [6–8].

In this paper, we propose a deep learning model for classifying folktales, and also for explaining predictions. By adapting different components from state-of-the-art deep learning architectures (e.g. ideas such as the Hierarchical Attention Network, the penalized hyperbolic tangent as an activation function, or the use of sparsemax as an alternative to softmax), this model shows improved performance for predicting ATU categories, compared to baselines adapted from previous research. Additionally, as far as we are aware, this is the first attempt to perform cross-lingual classification and visualization of predictions in folkloristics.

The rest of this document is organized as follows: Section 2 covers related work, analyzing previous relevant research. Section 3 describes the architecture of the model in detail. Section 4 details the datasets, the experimental methodology and results. Finally, Section 5 provides a conclusion to this endeavor, with final remarks and insights, as well as proposing options for future work.

2 Related Work

A classifier for folktales was developed and studied by Nguyen et al. [3], concerning works from the Dutch Folktale Database, which were organized according to the following narrative genres: *Fairy tales*, *Legends*, *Saint's legends*, *Urban legends*, *Personal narratives*, *Riddles*, *Situation puzzles*, *Jokes*, and *Songs*. To perform this task, the authors employed a support vector machine (SVM) with a linear kernel, L2 regularization, and the method by Crammer and Singer for multi-class classification [9], using features such as unigrams, character n-grams, and others. In the conducted experiments, the authors reported a macro-average F_1 score of 0.62 for classifying folktales according to their narrative genres, with ablation studies indicating a high impact from character n-grams.

In a following study, the authors developed a classifier using Learning to Rank and BM25 queries [4]. The same database was considered, but using labels from the ATU index and Type-Index of Urban Legends. The ranking of documents using BM25 explored two different baselines: the *big document model*, and the *small document model*, employing features such as information retrieval measures, lexical similarity, story similarity and subject-verb-object triplets. The results indicated a total mean reciprocal rank (MRR) accuracy of 0.82 for the ATU labels and 0.76 for sub-categories of Urban Legends.

More recently, research in similar NLP tasks has shown that other machine learning methods, e.g. deep neural networks, can generate better predictions. One such example is the Hierarchical Attention Network (HAN) method [7], which is inspired by the intrinsic structure of a document, where words form sentences, and sentences form documents. In this model, one bidirectional Gated Recurrent Unit (GRU) [10] encodes the words of each sentence into annotations, which are then weighted according to relevance by a softmax attention mechanism. Following this, a similar bidirectional GRU encodes sentences, which are also weighted by an identical mechanism, resulting in a representation of the whole document. Experiments on different text classification tasks showed consistent improvements compared to SVM baselines, and several other studies have since then applied similar models [8].

Additionally, the weights from the attention layers can be considered to generate a visualization of relevant features [6], which may be desirable to explain results. For example, by inspecting the attention weights in a given prediction, it is possible to generate text with important terms and phrases highlighted.

3 The Proposed Classification Method

In our research, we develop a model inspired by the HAN [7, 8], but modifying the attention layer to use the sparsemax function [11] and the activation functions, which are replaced by the penalized hyperbolic tangent [12]. We also extend it with a k-nearest neighbors (KNN) component [13], which is used to find and encode similar documents. Additionally, the proposed model considers two outputs for categories and subcategories, activated by the softmax function. To illustrate this architecture, a diagram is shown in Figure 1.

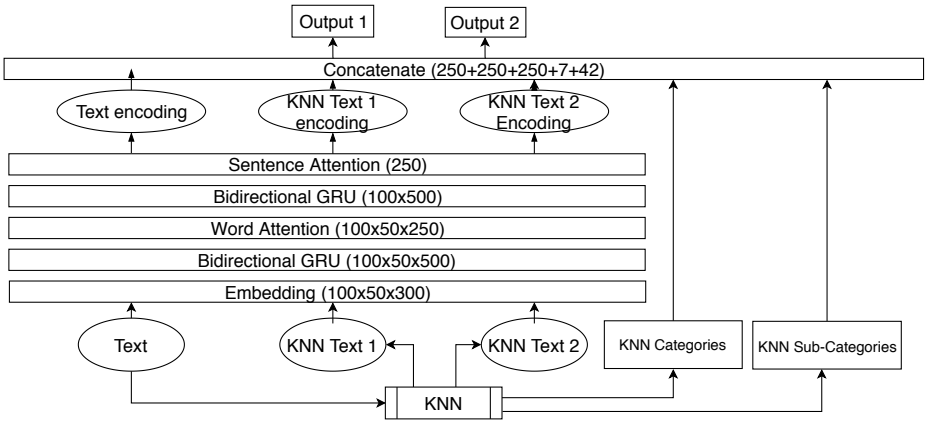

Fig. 1: The KNN augmented Hierarchical Attention Network.

In the remainder of this chapter, Section 3.1 covers the neural architecture. Section 3.2 provides details about the cross-lingual embeddings used in our approach. Finally, Section 3.3 provides some details concerning the KNN augmentation.

3.1 The Neural Architecture

In the proposed neural architecture, both sentence and document encoders are bidirectional GRUs, i.e. recurrent layers traditionally defined as:

$$\begin{aligned} s_j = \mathrm{R}_{\mathrm{GRU}}(s_{j-1}, x_j) &= (1-z)s_{j-1} + z \odot h \\ z &= \sigma(x_j W^{xz} + h_{j-1} W^{hz}) \\ r &= \sigma(x_j W^{xr} + h_{j-1} W^{hr}) \\ h &= \tanh(x_j W^{xh} + (h_{j-1} \odot r) W^{hg}) \\ y_j = \mathrm{O}_{\mathrm{GRU}}(s_j) &= s_j \end{aligned} \tag{1}$$

The reset gate r is used to access the previous state and computing an update to h, and the update gate z is used to compute the state s_j. To obtain further context, a bidirectional GRU is employed, concatenating sequences processed in forward and reverse order:

$$\begin{aligned} h_{x_{it}} &= \overrightarrow{h}_{it} + \overleftarrow{h}_{i(T-t)} \\ &= \overrightarrow{\mathrm{GRU}}(x_{it}) + \overleftarrow{\mathrm{GRU}}(x_{i(T-t)}) \end{aligned} \tag{2}$$

Algorithm 1 Sparsemax evaluation

Input: $\boldsymbol{z}$
Sort $\boldsymbol{z}$ as $z_{(1)} \geq \ldots \geq z_{(K)}$
Find $k(\boldsymbol{z}) := \max \left\{ k \in [K] \mid 1 + kz_{(k)} > \sum_{j \leq k} z_{(j)} \right\}$
Define $\tau(\boldsymbol{z}) = \frac{\left(\sum_{j \leq k(\boldsymbol{z})} z_{(j)} \right) - 1}{k(\boldsymbol{z})}$
Output: $\boldsymbol{p}$ s.t. $p_i = [z_i - \tau(\boldsymbol{z})]_+$.

Furthermore, instead of the traditional tanh activation function, we used the penalized hyperbolic tangent function [12], which has been shown to improve performance in a variety of NLP tasks. This function is defined as follows:

$$\text{penalized } \tanh(x) = \begin{cases} \tanh(x), & x > 0, \\ 0.25 \ \tanh(x), & x \leq 0 \end{cases} \tag{3}$$

Additionally, extracting and annotating the most significant terms is accomplished by using a word attention mechanism. This layer employs a feed forward network leveraging the sparsemax [11] activation function, and performs weighting such that the sum of all non-negative weights is exactly 1, similarly to a probabilistic distribution:

$$u_{it} = \text{penalized } \tanh(W_w h_{it} + b_w) \tag{4}$$
$$\alpha_{it} = \text{sparsemax}(u_{it}) \tag{5}$$
$$s_i = \alpha_{it} h_{it} \tag{6}$$

In the previous equation, u_{it} are the hidden representation resulting from a single layer MLP, a_{it} are the attention weights, and s_i is the new annotation corresponding to a single vector encoding an entire sequence.

The sparsemax function has been shown to improve predictions and visualizations. By generating distributions of attention weights with many zero values, it generates outputs with fewer, but more relevant, annotations. The procedure used for computing this function is shown in Algorithm 1.

3.2 Cross Language Word Embeddings

One challenge in our particular classification task is the multiplicity of languages in the MFTD dataset. In order to leverage training from different languages, cross-lingual embeddings from the MUSE project[3] were used.

[3] https://github.com/facebookresearch/MUSE

To generate these embeddings, the MUSE project took word vectors for different languages, specifically the FastText embeddings [14] trained with Wikipedia data, and then mapped them to match the English embeddings, such that a source word will hold similar values to its English translation. This aligning process was performed by employing the supervised iterative Procrustes method [15], which aims to find a transformation W that aligns embeddings X into Y:

$$W^* = \text{argmin}||WX - Y||_F = UV^T, \text{with } U\Sigma V^T = \text{SVD}(YX^T) \tag{7}$$

3.3 The KNN Augmentation

Inspired by previous research [13], a KNN component was developed to find documents similar to the input text. First, the two nearest neighbors from the training dataset are discovered by employing a KNN algorithm on texts represented as TF-IDF matrices of word unigrams. These neighbors are then encoded by the neural model, such that the concatenation of document encodings from both source and neighbor texts is used for predictions.

Furthermore, the labels and relative distances of neighbor texts are also considered. The labels are two one-hot encoded vector representing the categories and sub-categories of the neighbor texts. These are then multiplied by the distance from source s to neighbor n texts, which is computed by the Minkowski distance, defined as:

$$d = \frac{\sum_i |s_i - n_i|^p}{1/p} \tag{8}$$

where the order p is defined as 1. Finally, these are averaged across all neighbors, resulting in one feature for categories, and another for sub-categories.

4 Experimental Evaluation

To assess the quality of the proposed architecture, training and testing was performed with examples from the Multilingual Folk Tale Database (MFTD). Section 4.1 introduces the dataset, while Section 4.2 explains the evaluation protocol. The following Section 4.3 discusses the obtained results, and finally, Section 4.4 presents the visualizations of predictions.

4.1 The MFTD Dataset

The Multilingual Folk Tale Database (MFTD) is one of the largest online databases of folktales, containing thousands of texts in different languages. The documents used in our work were extracted from the MFTD website in order to build a dataset for training and testing, including labeled examples written in

Table 2: Statistics for the MFTD dataset concerning labeled examples.

Property	**EN**	**PT**	**ES**	**FR**	**IT**	**Total**
Number of examples	341	67	186	108	132	834
Sentences per text (average)	63.9	47.9	54.1	61.9	55.3	58.8
Sentences per text (90th percentile)	133.0	94.4	100.0	125.9	101.7	122.0
Words per sentence (average)	29.6	29.4	27.4	26.0	24.1	27.8
Words per sentence (90th percentile)	57.0	59.0	52.0	50.0	43.0	53.0
Size of vocabulary	15,617	9,739	20,329	13,343	13,642	72,670
Distribution of examples						
C_0: Animal tales	79	16	28	31	18	172
C_1: Tales of magic	175	43	96	56	70	440
C_2: Religious tales	8	1	14	5	8	36
C_3: Realistic tales	14	3	12	3	7	39
C_4: Tales of the stupid ogre	7	0	3	0	1	11
C_5: Anecdotes and jokes	46	3	27	11	23	110
C_6: Formula tales	7	1	6	2	5	26

English, Portuguese, Spanish, French and Italian. This resulted in a total of 834 examples, as further detailed in Table 2.

All the texts from the resulting dataset are classified according to the ATU system. Due to the scarce amount of examples for each individual index, the first two levels from this hierarchy were used as labels, resulting in a single-label, cross-lingual, multi-class dataset with 7 categories and 42 subcategories.

4.2 Evaluation Protocol

The experimental methodology used to assess the performance of the proposed classification model was based on cross-validation with a stratified k-fold partitioning algorithm using $k = 10$, and thus considering multiple splits of 90% and 10% for training and testing. This stratification was performed according to language-category labels, so that each split contains a balanced number of examples for a given pair of language and category. When comparing predictions and labels, the results were measured using micro and macro-averaged F_1 scores for ATU categories and sub-categories.

In terms of model parameters, namely the dimensionality of model components, the embedding layer was composed of 300 units, while the GRU and attention layers consisted of 250 units. In terms of features, matrices of 100x50 word identifiers were used, in proximate conformity with the 90th percentiles of the MFTD dataset. Finally, training was performed in batches of 20 instances, for a maximum of 50 epochs and using the categorical cross-entropy loss function and Adam optimization algorithm [16]. To develop and train this model, the Keras[4] and scikit-learn[5] libraries were used.

[4] https://keras.io/

[5] https://scikit-learn.org

Table 3: Evaluation results for the MFTD examples in English.

		Categories		Subcategories	
Model	**Features**	**Micro F_1**	**Macro F_1**	**Micro F_1**	**Macro F_1**
SVM	TF-IDF of word unigrams	0.710	0.282	**0.365**	0.233
SVM	TF-IDF of char [2,5]-grams	0.695	0.261	0.361	**0.235**
KNN	TF-IDF of word unigrams	0.581	0.188	0.193	0.099
KNN	TF-IDF of char [2,5]-grams	0.538	0.146	0.190	0.093
HAN	100x50 words, 250 units	**0.712**	**0.358**	0.245	0.149
KNN-HAN	100x50words, 250 units	0.648	0.313	0.179	0.090

Table 4: Evaluation results for the complete set of MFTD examples, in English, Portuguese, Spanish, French, and Italian.

		Categories		Subcategories	
Model	**Features**	**Micro F_1**	**Macro F_1**	**Micro F_1**	**Macro F_1**
HAN	100x50 word, 250 units	**0.780**	**0.487**	**0.396**	**0.277**
KNN-HAN	100x50 words, 250 units	0.768	0.471	0.290	0.170

To explain predictions, words from the source document were highlighted according to the attention weights of their corresponding annotations, while sentence attentions were displayed parallel to them.

4.3 Experimental Results

Initial experiments were conducted with the English subset of the data. As baselines, we used a linear SVM with L2 regularization, inspired by the model from previous studies on the classification of folktales [3], and a KNN classifier with k=2, which was also used in the augmented neural model. Regarding the neural models, both the HAN leveraging only text from the input document and the k-nearest neighbors augmented HAN (KNN-HAN) were used. As observed in Table 3, the neural models constantly outperformed the KNN baseline. Compared to the SVM, the base Hierarchical Attention Network achieves better performance regarding categories, but not for sub-categories. However, the augmentation in the model lead to a deterioration in performance.

With the complete multi-lingual dataset, we omit the baselines, as the differences in vocabulary for each language make them unsuited for cross-lingual classification. Once again, the augmented model did not show improvements for any metric when compared to the base HAN, as shown in Table 4. Details regarding the performance of each category are listed in Table 5, showing a tendency for worse performance in categories with fewer examples, namely C4, which had the least amount of texts. However, this tendency does not happen

Table 5: F_1 scores for each category.

Model	C0	C1	C2	C3	C4	C5	C6
HAN	**0.958**	**0.910**	**0.555**	**0.217**	0.000	**0.581**	0.233
KNN-HAN	0.941	0.906	0.415	0.201	**0.100**	0.562	**0.133**

Table 6: F_1 scores concerning categories for each language.

Model	Metric	English	Portuguese	Spanish	French	Italian
HAN	Micro F_1	**0.766**	0.764	**0.759**	**0.850**	**0.805**
KNN-HAN	Micro F_1	0.756	**0.793**	0.743	0.817	0.791
HAN	Macro F_1	**0.425**	0.572	0.509	**0.737**	**0.625**
KNN-HAN	Macro F_1	0.431	**0.660**	**0.510**	0.617	0.588

regarding performance across different languages (i.e., in some cases, languages with fewer examples end up achieving a higher performance), as shown in Table 6.

To study the impact of each feature in the cross-language classifier, ablation studies were performed by modifying components or hyper-parameters, as detailed in Table 7. These experiments indicate a favorable outcome from the sparsemax attention mechanism and choice of a higher dimensionality for the intermediate representations given by the neural layers, as performance decreases when modifying them. However, the penalized hyperbolic tangent seems to have only improved the prediction of categories. Lastly, the KNN augmentation (corresponding to the complete KNN-HAN architecture) resulted in deterioration of performance for all metrics, showing it did not succeed at improving the model.

4.4 Visualization of Results

Results can be explained, to some extent, by the relevance attributed to words and sentences during a prediction, i.e. the attention weights associated with their representations. As such, we can use these values to generate a visualization which displays the weight attributed to each word or sentence, e.g. by highlighting

Table 7: Ablation experiments for the HAN with multi-lingual examples from the MFTD dataset.

	Categories		Subcategories	
Model	**Micro F_1**	**Macro F_1**	**Micro F_1**	**Macro F_1**
HAN	0.780	0.487	0.396	0.277
Softmax Attention	0.748	0.422	0.363	0.236
Tanh Activation	0.764	0.449	0.410	0.282
KNN Augmentation	0.768	0.471	0.290	0.170
150 Units in Neural Layers	0.740	0.443	0.328	0.212

Table 8: The 20 words with highest total attention per category.

Category	**Most relevant words**
Animal tales	fox, wolf, cat, mouse, dog, man, ass, sparrow, lion, cock, turtle, hare, wood, bird, bear, master, goose, he, sheep, hen
Tales of magic	king, man, princess, daughter, lad, mother, father, girl, he, prince, she, old, woman, brothers, castle, son, child, bird, boy, brother
Religious tales	lord, king, father, tailor, shepherd, son, he, man, boy, peter, brother, merchant, woman, child, master, peasant, mother, saint, sword, flute
Realistic tales	king, princess, bride, prince, soldier, woman, maid, daughter, she, sweetheart, girl, father, bird, old, huntsman, son, man, beggar, lion, he
Stupid ogre	troll, tailor, lad, giant, peasant, devil, wood, old, maiden, woman, shepherd, bear, girl, boots, witch, man, child, roland, he, she
Jokes	man, king, lad, mother, cow, peasant, goody, hans, he, master, farmer, tailor, claus, boy, horse, old, goose, woman, frederick, she
Formula tales	cock, hen, mother, fox, pancake, sister, hare, little, pig, father, bride, tup, henny, horse, girl, dogs, woodcutter, shoes, birds, tree

words. Using colors represented in the Hue Saturation Lightness (HSL) scale, the lightness value can be calculated as $L = 1 - C.w_i$ to generate darker highlights for higher attention weights w_i, scaled by C to improve visual contrast.

From the obtained results for the overall corpus, we observe a prevalence of higher attention weights for words related to the concepts of *animals* (e.g. goat, wolf) and *royalty* (e.g. king, princess). To study differences across categories, the sum of attention values for identical terms was considered for the English subset, and the words with highest values listed in Table 8. This listing demonstrates an intuitive connection between *Animal tales* and its most relevant words which is distinct from other categories, while the relevant vocabulary for categories such as *Tales of magic*, *Religious tales* and *Realistic tales* exhibits many similarities.

For other languages, the correspondent translations are often attributed similar weights, as exemplified in Figure 2.

Concerning sentence attention, while it is harder to analyze their impact, higher attention values are often attributed to sentences containing words which are more relevant to intuition.

5 Conclusions and Future Work

In this work, we modified and extended the Hierarchical Attention Network (HAN) to classify folktales and generate visualizations which aid explaining results.

Concerning an English subset of MFTD folktales, the baseline HAN model achieved the best performance for classifying categories, outperforming baselines inspired by previous research, but not for sub-categories. This might be explained due to the low number of examples present in most sub-categories, which is a common bottleneck for the training of neural networks. The model augmented with a KNN component did not show improvements for any metrics. Regarding the multi-lingual dataset, once again, the extended model did now show improvements of results, performing worse than the base HAN architecture.

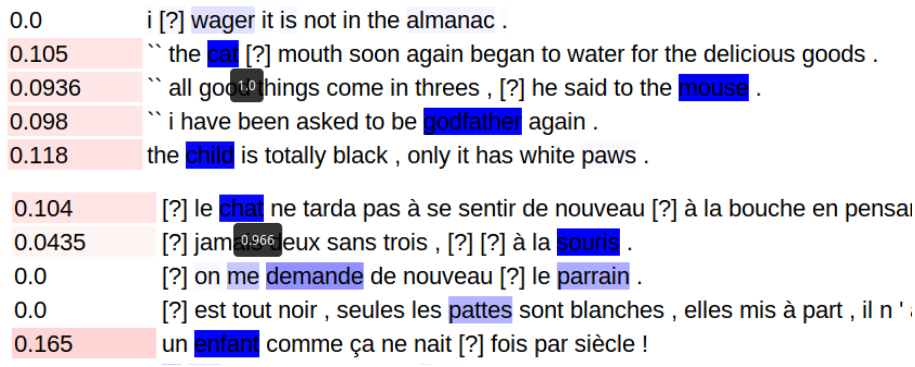

Fig. 2: Highlighted excerpts for the English and French versions of of the *Cat and Mouse Partnership* tale, originally written by the Grimm brothers.

By performing ablation studies with the multi-lingual dataset, we noticed that the sparsemax attention mechanism showed the most impact, improving both classifications and visualizations. There was also a superior performance when increasing the dimensionality used in the intermediate representations, while mixed results were obtained for the penalized hyperbolic tangent activation function, which improved predictions of categories but not of sub-categories. Concerning the KNN augmentation, it was not successful at improving the base architecture, showing a negative impact for all metrics.

Since performance tended to scale with the number of examples for a given category, this suggests that results would improve from more training data, namely on classes with fewer texts. On the other hand, even though the cross-lingual embeddings seem effective at aligning terms to their English translation, performance varied for each language, independently of the number or distribution of examples, possibly due to differences in the structure of each language.

For future work, using pre-trained contextual representations [17] is an option which might improve results. There has also been recent research considering the hierarchical document structure [18, 19], which may be considered to improve results. Additionally, other neural architectures could also be explored, such as the Transformer model [20], which is based solely on attention mechanisms. Finally, although this was not explored, the KNN mechanism could be used to improve the interpretability of results, e.g. by providing a list of neighbor texts.

References

1. James Abello, Peter Broadwell, and Timothy R. Tangherlini. Computational folkloristics. *Communications of the ACM*, 55(7), 2012.
2. Theo Meder, Folgert Karsdorp, Dong Nguyen, Mariët Theune, Dolf Trieschnigg, and Iwe Muiser. Automatic enrichment and classification of folktales in the Dutch folktale database. *Journal of American Folklore*, 129(511), 2016.

3. Dong Nguyen, Dolf Trieschnigg, Theo Meder, and Mariët Theune. Automatic classification of folk narrative genres. In *Proceedings of the Workshop on Language Technology for Historical Text*, 2012.
4. Dong Nguyen, Dolf Trieschnigg, and Mariet Theune. Folktale classification using learning to rank. In *Proceedings of the European Conference on Information Retrieval*, 2013.
5. Yoav Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57, 2016.
6. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
7. Zichao Yang, Diyi Yan, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2016.
8. Francisco Duarte, Bruno Martins, Cátia Sousa Pinto, and Mário J Silva. Deep neural models for ICD-10 coding of death certificates and autopsy reports in free-text. *Journal of Biomedical Informatics*, 80, 2018.
9. Koby Crammer and Yoram Singer. On the learnability and design of output codes for multiclass problems. *Machine learning*, 47(2-3).
10. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
11. Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the International Conference on Machine Learning*, 2016.
12. Steffen Eger, Paul Youssef, and Iryna Gurevych. Is it time to swish? comparing deep learning activation functions across NLP tasks. *arXiv preprint arXiv:1901.02671*, 2019.
13. Zhiguo Wang, Wael Hamza, and Linfeng Song. k-nearest neighbor augmented neural networks for text classification. *arXiv preprint arXiv:1708.07863*, 2017.
14. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
15. Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.
16. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
17. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
18. Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. End-to-end hierarchical text classification with label assignment policy. In *International Conference on Learning Representations*, 2019.
19. Simon Baker and Anna Korhonen. Initializing neural networks for hierarchical multi-label text classification. In *Proceedings of Workshop on Biomedical Natural Language Processing*.
20. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Annual conference on Neural Information Processing Systems*.