Building on Word Animacy to Determine Coreference Chain Animacy in Cultural Narratives

Labiba Jahan, Geeticka Chauhan, Mark A. Finlayson
School of Computing and Information Sciences
Florida International University
Miami, FL 33199
{ljaha002, gchau005, markaf}@fiu.edu

Abstract

Animacy is the characteristic of being able to independently carry out actions in a story world (e.g., movement, communication). It is a necessary property of characters in stories, and so detecting animacy is a useful step in automatic story understanding. Prior approaches to animacy detection have conceived of animacy as a word- or phrase-level property, without explicitly connecting it to characters. In this work we compute the animacy of referring expressions directly using a statistical approach incorporating useful features. We then compute the animacy of coreference chains via a majority vote of the animacy of the chain's constituent referring expressions. We also reimplement prior approaches to word-level animacy to compare performance. We demonstrate these results on a small set of folktales with gold-standard annotations for coreference structure and animacy (15 Russian folktales translated into English). We achieve an F_1 measure 0.90 for the referring expression animacy model, and 0.86 for the coreference chain model.

1 Introduction

Characters are an indispensable element of narrative. Most definitions of narrative acknowledge the central role of character: Monika Fludernik, as just one example of many, defines a narrative as "a representation of a possible world . . . at whose centre there are *one or several protagonists* of an anthropomorphic nature . . . who (mostly) perform goal-directed actions . . . " (2009, p. 6). Thus, if we are to achieve the long-term goal of automatic story understanding, we need to identify a story's characters.

One subtask of character detection is *animacy* detection, where animacy is the characteristic of being able to independently carry out actions in a story world (e.g., movement or communication). All characters are necessarily animate—although not all animate things are necessarily characters—and so detecting animacy will immediately narrow the set of possibilities for character detection.

Some theorists have proposed closed lists of linguistic expressions that should be automatically considered to indicate animate entities, such as titles, animals, or personal pronouns (e.g., Quirk et al., 1985; Yamamoto, 1999). However, stories can arbitrarily introduce characters that would not be animate in real life, for example, walking stoves or talking trees.

Prior work has conceived of animacy as a word-level phenomenon, marking animacy as an independent feature on each individual word (e.g., Orăsan and Evans, 2007; Bowman and Chopra, 2012; Karsdorp et al., 2015). However, characters and other entities are expressed in texts as coreference chains made up of referring expressions (Jurafsky and Martin, 2007), and so we need some way of computing animacy on the chain themselves. Here we take the approach of computing animacy directly on referring expressions themselves and then use majority vote of referring expression-level animacy to compute animacy of coreference chains.

2 Data

No prior data in English was readily available to use for our work. Orăsan and Evans (2007) did their work in English but their data was not readily available.

Our data was a small corpus of 15 Russian folktales that we assembled in the context of other work (Finlayson, 2017). The corpus contains old-standard annotations for token and sentence boundaries, parts of speech (Penn Treebank II Tagset, Marcus et al., 1993), referring expressions, and coreference chains (as well as other layers of annotation).

We annotated these tales for coreference- and word-level animacy. Agreement for the word-level is 0.97 Cohen's kappa (κ) and for the coreference-level is 0.99 κ, which represents near-perfect overall agreement (Landis and Koch, 1977). The animacy of referring expressions were directly calculated from the animacy of the coreference chains.

We also annotated every word in the corpus for animacy directly. We marked as animate all nouns, gendered pronouns and adjectives that would refer to animate entities in real life (such humans or animals, as discussed in Quirk et al., 1985). We also marked as animate any words directly referring to entities that acted animately in a story, regardless of the default inanimacy of the words. Examples of animate and inanimate expressions are given in Table 1.

3 Experimental Setup

3.1 Features

1. **Word Embeddings (WE)**: We computed word embeddings in 300 dimensions for all the words in the stories using the skip-gram architecture algorithm (Mikolov et al., 2013) and DeepLearning4J library (Deeplearning4j Development Team, 2017). This is a vector feature drawn from Karsdorp et al. (2015), and is primarily relevant to classifying word-level animacy.

2. **Word Embeddings on Ref. Exp. (WER)**: We calculated word embeddings as a vector feature in 450 dimensions for just the words within the referring expressions, again using the skip-gram approach.

3. **Composite Word Embedding (CWE)**: We computed a composite word embedding for the neighborhood of each word, adding together the word embedding vectors for three words before and three words after the target word (excluding the target). This is also a vector feature, and is again partially drawn from Karsdorp et al. (2015). The idea of this feature is that it estimates the similarities of the context among all animate words (or all inanimate words).

4. **Parts of Speech (POS)**: By analogy with the other embeddings, we computed an embedding over part of speech tags in 300 dimensions, with the same settings as in feature #1 (WE). This feature models the tendency of nouns, pronouns, and adjectives to refer to animate entities.

5. **Noun (N)**: We checked whether a given referring expression contained a noun, and encoded this as a boolean feature. This feature explicitly captures the tendency of nouns to refer to animate entities.

6. **Grammatical Subject (GS)**: Animate references tend to appear as the grammatical subjects of verbs (Ovrelid, 2005). We used dependency parses generated by the Stanford dependency parser (Manning et al., 2014) to check if a given referring expression was used as a grammatical subject relative to any verb in the sentence, and encoded this as a boolean feature.

7. **Semantic Subject (SS)**: We also computed whether or not a referring expression appeared as a semantic subject (ARG0) to a verb. We used the semantic role labeler associated with the Story Workbench annotation tool (Finlayson, 2008, 2011) to compute semantic roles for all the verbs in the stories.

3.2 Classification Models

We implemented our classification models using SVM (Chang and Lin, 2011), with a radial basis function kernel. We trained each model using cross validation, and report macroaverages across the performance on test folds.

We constructed three models for animacy: referring expressions, coreference chains, and a reimplentation for words. For our referring expression animacy model, we explored different combinations of the features: word embedding over referring expressions (WER), noun (N), grammatical subject (GS), and semantic subject (SS). We configured the SVM with $\gamma = 1$, $C = 0.5$ and $p = 1$. We measured the performance of the classifier using 10-fold cross validation.

We calculated two baselines for referring expression animacy. The first is the majority class baseline (inanimate is the majority class). The second combines word-level animacy predictions generated by our word animacy model (discussed below) via a majority vote.

For the coreference chain animacy model, we implemented two majority vote approaches for combining the results of the referring expression animacy model to obtain a coreference animacy prediction. First, we computed the majority vote considering all referring expressions in a coreference chain. Because short coreference chains were responsible for much of the poor performance, we also calculated the performance of majority voting excluding chains of length four and below.

To compare with prior work, we also implemented a word animacy model, adapting an exist-

Referring Expression	Class	Explanation
a princess, the dragon, the tsar	Animate	Normally animate entities
walking stove, talking tree	Animate	inanimate entities but are animate in context
Kiev, this world, every house	Inanimate	Normally inanimate objects
Word		
princess, dragon, he, she	Animate	Nouns and pronouns denoting animate entities
kind [princess], stronger [dragon]	Animate	Adjectives that suggest animacy
it, that, this	Inanimate	Personal pronouns referring to inanimate objects

Table 1: Examples of annotation of coreference- and word-level animacy.

ing system with the best performance (Karsdorp et al., 2015). That model used features based on word N-grams, parts of speech, and word embeddings. Similarly, we implemented our classifier using word embeddings over words (WE), combined word embeddings (CWE), and parts of speech (POS). The SVM was configured with $\gamma = 5$, $C = 5000$ and $p = 1$, and we measured the performance with 20-fold cross validation.

4 Results & Discussion

We obtained the best result (F_1 of 0.90) for our referring expression model using three features: word embeddings over referring expressions (WER), noun (N) and semantic subject (SS). For the coreference animacy model, majority vote does not work as well as expected, with an overall F_1 of 0.61 when calculated over all chains but we got F_1 of 0.86 when we calculated over long chains (those with more than four referring expressions) only. This suggests that in future work we need to concentrate our effort on solving the short chain issue. Finally, our word model achieved F_1 of 0.90 where the state of the art achieved F_1 of 0.93.

5 Error Analysis & Future Work

Determining the animacy of short coreference chains is apparently a challenging task for our system. We believe one approach to solving this problem is more data. The second problem is that many quotes are full of animate words. This will require some rule-based processing to address. Finally, in the folktales we see names whose surface form are identical to inanimate entities. Addressing this will requiring integrating named entity recognition into the system.

6 Related Work

6.1 Animacy Detection in English

Evans and Orăsan (2000) first explored animacy classification as a means to improve anaphora resolution. They took this work forward by using a supervised machine learning (ML) method to mark unseen WordNet senses by their animacy (Orăsan and Evans, 2001). They also explored both rule-based and machine-learning-based for animacy classification of nouns (Orăsan and Evans, 2007). Bowman and Chopra (2012) conceived of animacy and inanimacy classification as a multi-class problem applied directly to noun phrases (NPs), using a maximum entropy classifier to classify NPs as *human*, *vehicle*, *time*, *animal*, etc, with an overall accuracy of 85%.

6.2 Animacy Detection in Other Languages

Nøklestad (2009) implemented animacy detection for Norwegian nouns, levering this along with Named Entity Recognition (NER) to improve the performance of anaphora resolution. This method achieves an accuracy of 93%. Bloem and Bouma (2013) developed an automatic animacy classifier for Dutch nouns, by dividing them into *Human*, *Nonhuman* and *Inanimate* classes. Prediction of the *Human* achieved 87% accuracy, and the large inanimate class was predicted correctly 98% of the time. Karsdorp et al. (2015) developed a word-level animacy model for Dutch, tested on Dutch folktales.

7 Acknowledgments

This work was partially supported by Office of Naval Research (ONR) grant number N00014-17-1-2983.

References

Jelke Bloem and Gosse Bouma. 2013. Automatic animacy classification for Dutch. *Computational Linguistics in the Netherlands Journal (CLIN)*, 3:82–102.

Samuel R Bowman and Harshit Chopra. 2012. Automatic animacy classification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT): Student Research Workshop*, pages 7–10. Montreal, Canada.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Deeplearning4j Development Team. 2017. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org. Accessed: 2017-04-08.

Richard Evans and Constantin Orăsan. 2000. Improving anaphora resolution by identifying animate entities in texts. In *Proceedings of the Discourse Anaphora and Reference Resolution Conference (DAARC2000)*, pages 154–162. Lancaster, England.

Mark A Finlayson. 2011. The Story Workbench: An extensible semi-automatic text annotation tool. In *Proceedings of the 4th Workshop on Intelligent Narrative Technologies (INT4)*, pages 21–24. Stanford, CA.

Mark A. Finlayson. 2017. ProppLearner: Deeply Annotating a Corpus of Russian Folktales to Enable the Machine Learning of a Russian Formalist Theory. *Digital Scholarship in the Humanities*, 32(2):284–300.

Mark Alan Finlayson. 2008. Collecting semantics in the wild: The story workbench. In *Proceedings of the AAAI Fall Symposium on Naturally Inspired Artificial Intelligence*, pages 46–53. Arlington, VA.

Monika Fludernik. 2009. *An Introduction to Narratology*. Routledge, New York.

Daniel Jurafsky and James H. Martin. 2007. *Speech and Language Processing*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Folgert Karsdorp, M Meulen, Theo Meder, and APJ van den Bosch. 2015. Animacy detection in stories. In *Proceedings of the 6th Workshop on Computational Models of Narrative (CMN'15)*, pages 82–97. Atlanta, GA.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 55–60. Baltimore, MD.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. http://arxiv.org/abs/1301.3781.

Anders Nøklestad. 2009. *A Machine Learning Approach to Anaphora Resolution Including Named Entity Recognition, PP Attachment Disambiguation, and Animacy Detection*. Ph.D. thesis, University of Oslo, Oslo, Norway.

Constantin Orăsan and Richard Evans. 2001. Learning to identify animate references. In *Proceedings of the 2001 Workshop on Computational Natural Language Learning (CoNLL)*, page Article No. 16. Toulouse, France.

Constantin Orăsan and Richard J Evans. 2007. NP animacy identification for anaphora resolution. *Journal of Artificial Intelligence Research*, 29(1):79–103.

Lilja Ovrelid. 2005. Animacy classification based on morphosyntactic corpus frequencies: some experiments with Norwegian nouns. In *Proceedings of the Workshop on Exploring Syntactically Annotated Corpora*, pages 24–34. Birmingham, England.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, London.

Mutsumi Yamamoto. 1999. *Animacy and reference: A cognitive approach to corpus linguistics*. John Benjamins Publishing, Amsterdam.