Animacy Detection with Voting Models

Joshua L. Moore*
Dept. Computer Science
Cornell University
Ithaca, NY 14853
jlmo@cs.cornell.edu

Christopher J.C. Burges Erin Renshaw Wen-tau Yih
Microsoft Research
One Microsoft Way
Redmond, WA 98052
{cburges, erinren, scottyih}
@microsoft.com

Abstract

Animacy detection is a problem whose solution has been shown to be beneficial for a number of syntactic and semantic tasks. We present a state-of-the-art system for this task which uses a number of simple classifiers with heterogeneous data sources in a voting scheme. We show how this framework can give us direct insight into the behavior of the system, allowing us to more easily diagnose sources of error.

1 Introduction

Animacy detection has proven useful for a variety of syntactic and semantic tasks, such as anaphora and coreference resolution (Orăsan and Evans, 2007; Lee et al., 2013), verb argument disambiguation (Dell'Orletta et al., 2005) and dependency parsing (Øvrelid and Nivre, 2007). Existing approaches for animacy detection typically rely on two types of information: linguistic databases, and syntactic cues observed from the corpus. They usually combine two types of approaches: rule based systems, and machine learning techniques. In this paper we explore a slightly different angle: we wish to design an animacy detector whose decisions are *interpretable and correctable*, so that downstream semantic modeling systems can revisit those decisions as needed. Thus here, we avoid defining a large number of features and then using a machine learning method such as boosted trees, since such methods, although powerful, result in hard-to-interpret systems. Instead, we explore combining interpretable voting models using machine learning only to reweight their votes. We show that such an approach can indeed result in a high performing system, with animacy detection accuracies in the mid 90% range, which compares well with other reported rates. Ensemble methods are well known (see for example, Dietterich (2000)) but our focus here is on using them for interpretability while still maintaining accuracy.

* Work performed while visiting Microsoft Research.

2 Previous Work

2.1 Definitions of Animacy

Previous work uses several different definitions of animacy. Orăsan and Evans (2007) define animacy in the service of anaphora resolution: an NP is considered animate "*if its referent can also be referred to using one of the pronouns he, she, him, her, his, hers, himself, herself, or a combination of such pronouns (e.g. his/her)*". Although useful for the task at hand, this has counterintuitive consequences: for example, *baby* may be considered animate or inanimate, and *ant* is considered inanimate (Ibid., Figure 1). Others have argued that animacy should be captured by a hierarchy or by categories (Aissen, 2003; Silverstein, 1986). For instance, Zaenen et al. (2004) propose three levels of animacy (*human*, *other animate* and *inanimate*), which cover ten categories of noun phrases, with categories like ORG (organization), ANIM (animal) and MAC (intelligent machines such as robots) categorised as *other animate*. Bowman and Chopra (2012) report results for animacy defined both this way and with the categories collapsed to a binary (*animate, inanimate*) definition.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 55–60,
Seattle, Washington, USA, 18-21 October 2013.

2.2 Methods for Animacy Detection

Evans and Orăsan (2000) propose a rule-based system based on the WordNet taxonomy (Fellbaum, 1998). Each synset is ascribed a binary animacy label based on its unique beginner. A given noun is then associated with the fraction of its animate synsets (where all synsets are taken to be animate or inanimate) and one minus that fraction, similarly for a given verb. Animacy is then ascribed by applying a series of rules imposing thresholds on those fractions, together with rules (and a gazetteer) to detect names and acronyms, and a rule triggered by the occurrence of *who*, or reflexives, in the NP. In later work, Orăsan and Evans (2007) extend the algorithm by propagating animacy labels in the WordNet graph using a chi-squared test, and then apply a k-nearest neighbor classifier based on four lexical features. In their work, the only context used was the animacy of the verb in the NP, for heads of subject NPs (e.g., the subject of *eat* is typically animate). Øvrelid (2009) and Bowman and Chopra (2012) extend this idea by using dependency relations to generate features for their classifier, enabled by corpora created by Zaenen et al. (2004). In another approach, Ji and Lin (2009) apply a simple "relative-pronoun" pattern to the Google n-gram corpus (Brants and Franz, 2006) to assign animacy (see the List model in Section 5 for details). Although the animacy decision is again context-independent, such a list provides a strong baseline and thus benefit applications like anaphora resolution (Lee et al., 2013).

3 The Task

We adopt a definition of animacy closest to the binary version in Bowman and Chopra (2012): we define an entity to be animate if it is alive and has the ability to move under its own will. We adopt this simple definition because it fits well with the common meaning and is therefore less error prone, both in terms of incorporation into higher level models, and for labeling (Orăsan and Evans (2007) report that the labeling of animacy tuned for anaphora proved challenging for the judges). We also apply the label to single noun tokens where possible: the only exceptions are compound names ("*Sarah Jones*") which are treated as single units. Thus, for example, "*puppy food*" is treated as two words, with *puppy* animate and *food* inanimate. A more complete definition would extend this to all noun phrases, so that *puppy food* as a unit would be inanimate, a notion we plan to revisit in future work. Note that even this simple definition presents challenges, so that a binary label must be applied depending on the predominant meaning. In "*A plate of chicken*," *chicken* is treated as inanimate since it refers to food. In "*Caruso (1873-1921) is considered one of the world's best opera singers. He...*," although at the time of writing clearly *Caruso* was not alive, the token is still treated as animate here because the subsequent writing refers to a live person.

4 The Data

We used the MC160 dataset, which is a subset of the MCTest dataset and which is composed of 160 grade level reading comprehension stories generated using crowd sourcing (Richardson et al., 2013). Workers were asked to write a short story (typically less than 300 words) with a target audience of 5 to 7 year olds. The available vocabulary was limited to approximately 8000 words, to model the reading ability of a first or second grader. We labeled this data for animacy using the definition given above. The first 100 of the 160 stories were used as the training set, and the remaining 60 were used for the test set. These animacy labels will be made available on the web site for MCTest (Richardson et al., 2013).

5 The Models

Since one of our key goals is interpretability we chose to use an ensemble of simple voting models. Each model is able to vote for the categories *Animal*, *Person*, *Inanimate*, or to abstain. The distinction between *Animal* and *Person* is only used when we combine votes, where *Animal* and *Person* votes appear as distinct inputs for the final voting combination model. Some voters do not distinguish between *Person* and *Animal*, and vote for *Animate* or *Inanimate*. Our models are:

List: The n-gram list method from (Ji and Lin, 2009). Here, the frequencies with which the relative pronouns *who*, *where*, *when*, and *which* occur are considered. Any noun followed most frequently by *who* is classified as *Animate*, and any other noun

in the list is classified as *Inanimate*. This voter abstains when the noun is not present in the list.

Anaphora Design: The WordNet-based approach of Evans and Orăsan (2000).

WordNet: A simple approach using WordNet. This voter chooses *Animal* or *Person* if the unique beginner of the first synset of the noun is either of these, and *Inanimate* otherwise.

WordSim: This voter uses the contextual vector space model of Yih and Qazvinian (2012) computed using Wikipedia and LA Times data. It uses short lists of hand-chosen signal words for the categories *Animal*, *Person*, and *Inanimate* to produce a "response" of the word to each category. This response is equal to the maximum cosine similarity in the vector space of the query word to any signal word in the category. The final vote goes to the category with the highest response.

Name: We used an in-house named entity tagger. This voter can recognize some inanimate entities such as cities, but does not distinguish between people and animals, and so can only vote *Animate*, *Inanimate* or *Abstain*.

Dictionaries: We use three different dictionary sources (Simple English Wiktionary, Full English Wiktionary, and the definitions found in WordNet) with a recursive dictionary crawling algorithm. First, we fetch the first definition of the query and use a dependency tree and simple heuristics to find the head noun of the definition, ignoring qualification NPs like "piece" or "member." If this noun belongs to a list of per-category signal words, the voter stops and votes for that category. Otherwise, the voter recursively runs on the found head noun. To prevent cycling, if no prediction is made after 10 recursive lookups, the voter abstains.

Transfer: For each story, we first process each sentence and detect instances of the patterns *x am/is/was/are/were y* and *y named x*. In each of these cases, we use majority vote of the remaining voters to predict the animacy of *y* and transfer its vote to *x*, applying this label (as a vote) to all instances of *x* in the text.

The *WordSim* and *Dictionaries* voters share lists of signal words, which were chosen early in the experimental process using the training set. The signal words for the *Animal* category were *animal* and *mammal*[1]. *Person* contains *person* and *people*. Finally, *Inanimate* uses *thing*, *object*, *space*, *place*, *symbol*, *food*, *structure*, *sound*, *measure*, and *unit*.

We considered two methods for combining voters: majority voting (where the reliable *Name* voter overrides the others if it does not abstain) and a linear reweighting of votes. In the reweighting method, a feature vector is formed from the votes. Except for *WordSim*, this vector is an indicator vector of the vote – either *Animal*, *Person*, *Animate* (if the voter doesn't distinguish between animals and people), *Inanimate*, or *Abstain*.

For *Dictionaries*, the vector's non-zero component is multiplied by the number of remaining allowed recursive calls that can be performed, plus one (so that a success on the final lookup gives a 1). For example, if the third lookup finds a signal word and chooses *Animal*, then the component corresponding to *Animal* will have a value of 9.

For *WordSim*, instead of an indicator vector, the responses to each category are used, or an indicator for abstain if the model does not contain the word. If the word is in the model, a second vector is appended containing the ratio of the maximum response to the second-largest response in the component for the maximum response category. These per-voter feature vectors are concatenated to form a 35 dimensional vector, and a linear SVM is trained to obtain the weights for combining the votes.

6 Results

We used the POS tagger in MSR SPLAT (Quirk et al., 2012) to extract nouns from the stories in the MC160 dataset and used these as labeled examples for the SVM. This resulted in 5,120 extracted nouns in the 100 training stories and 3,009 in the 60 test stories. We use five-fold cross-validation on the training set to select the SVM parameters. 57.2% of the training examples were inanimate, as were 58.1% of the test examples.

Table 1 gives the test accuracy of each voter. *List*

[1]This was found to work well given typical dictionary definitions despite the fact that people are also mammals.

List	Anaphora	WNet	WSim	Dict	Name
84.6	77.1	78.8	57.6	74.3	16.0

Table 1: Accuracy of various individual voters on the test set. Abstentions are counted as errors. Note that *Transfer* depends on a secondary source for classification, and is therefore not listed here.

	Majority	SVM
N+WN+D+WS+AD+L	87.7	95.0
N+WN+WS	80.1	95.0
N+WN+D+WS+AD+L+T	87.4	95.0
N+WN+D+WS	86.4	94.8
N+WN+WS+AD+L	86.5	94.7
N+WN+D+WS+T	86.8	94.0
N+WN+D	86.1	93.7
N+WN	89.3	93.0
N+D	82.6	93.0
N+AD	87.6	89.4
N+L	85.4	88.9

Table 2: Accuracy of various combinations of voters among Name (N), Anaphora Design (AD), List (L), WordNet (WN), WordSim (WS), Dictionary (D), and Transfer (T) under majority voting and SVM schemes. Bold indicates a statistically significant difference over the next lower bolded entry with $p < 0.01$, for the SVM.

comes out on top when taken alone, but we see in later results that it is less critical when used with other voters. *Name* performs poorly on its own, but later we will see that it is a very accurate voter which frequently abstains.

Table 2 gives the test performance of various combinations of voters, both under majority vote and reweighting. Statistical significance was tested using a paired t-test, and bold indicates a method was significant over the next lower bold line with p value $p < 0.01$. We see a very large gain from the SVM reweighting: 14.9 points in the case of *Name+WordNet+WordSim*.

In Table 3, we show the results of ablation experiments on the voters. We see that the most valuable sources of information are *WordSim* and *Dictionaries*.

Finally, in Table 4, we show a breakdown of which voters cause the most errors, for the majority vote system. In this table, we considered only "final errors," i.e. errors that the entire system makes. Over all such errors, we counted the number of times

	Majority	SVM
WordSim	87.6	93.7
SimpleWikt (dict)	87.3	94.1
FullWikt (dict)	86.4	94.3
Dict	87.4	94.5
Name	86.6	94.7
List	86.4	94.8
WordNet (dict)	88.7	94.8
WordNet	87.5	94.9
Anaphora Design	88.6	94.9
Transfer	87.7	95.0

Table 3: Test accuracy when leaving out various voters, using both majority vote and and reweighting. Bold indicates statistical significance over the next lower bold line with $p < 0.01$.

each voter chose incorrectly, giving a count of how many times each voter contributed to a final error. We see that the *Anaphora Design* system has the largest number of errors on both train and test sets. After this, *WordNet*, *List*, and *WordNet (dict)* are also large sources of error. On the other hand, *Name* and *WordSim* have very few errors, indicating high reliability. The table also gives the number of *critical errors*, where the voter selected the wrong category *and* was a deciding vote (that is, when changing its vote would have resulted in a correct overall classification). We see a similar pattern here, with *Anaphora Design* causing the most errors and *WordSim* and *Name* among the most reliable. We included *Anaphora Design* even though it uses a different definition of animacy, to determine if its vote was nevertheless valuable.

Error tables such as these show how voting models are more interpretable *and therefore correctable* compared to more complex learned models. The tables indicate the largest sources of error and suggest changes that could be made to increase accuracy. For example, we could make significant gains by improving *WordNet*, *WordNet (dictionary)*, or *List*, whereas there is relatively little reason to adjust *WordSim* or *Name*.

7 Conclusions

We have shown that linear combinations of voting models can give animacy detection rates in the mid 90% range. This is well above the accuracy found

	Errors		Critical	
	Train	Test	Train	Test
Anaphora Design	555	266	117	76
WordNet	480	228	50	45
List	435	195	94	45
Transfer	410	237	54	58
WordNet (dict)	385	194	84	65
SimpleWikt (dict)	175	111	39	16
FullWikt (dict)	158	67	1	5
WordSim	107	89	11	19
Name	71	55	27	19

Table 4: *Errors* column: number of errors on train and test where a source voted incorrectly, and was thus at least in part responsible for an error of the overall system. *Critical* column: number of errors on train and test where a source voted incorrectly, and in addition cast a deciding vote. Results are for majority vote.

by using the n-gram method of (Ji and Lin, 2009), which is used as an animacy detection component in other systems. In this sense the work presented here improves upon the state of the art, but there are caveats, since other workers define animacy differently and so a direct comparison with their work is not possible. Our method has the added advantage of interpretability, which we believe will be useful when using it as a component in a larger system.

Acknowledgments

We wish to thank Andrzej Pastusiak for his help with the labeling tool.

References

Judith Aissen. 2003. Differential object marking: Iconicity vs. economy. *Natural Language & Linguistic Theory*, 21(3):435–483.

Samuel Bowman and Harshit Chopra. 2012. Automatic animacy classification. In *Proceedings of the NAACL-HLT 2012 Student Research Workshop*.

Thorsten Brants and Alex Franz. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium.

Felice Dell'Orletta, Alessandro Lenci, Simonetta Montemagni, and Vito Pirrelli. 2005. Climbing the path to grammar: a maximum entropy model of subject/object learning. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, PMHLA '05, pages 72–81, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15.

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.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Heng Ji and Dekang Lin. 2009. Gender and animacy knowledge discovery from Web-scale n-grams for unsupervised person mention detection. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pages 220–229, Hong Kong, December. City University of Hong Kong.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4).

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

Lilja Øvrelid and Joakim Nivre. 2007. When word order and part-of-speech tags are not enough – Swedish dependency parsing with rich linguistic features. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 447–451.

Lilja Øvrelid. 2009. Empirical evaluations of animacy annotation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Chris Quirk, Pallavi Choudhury, Jianfeng Gao, Hisami Suzuki, Kristina Toutanova, Michael Gamon, Wen-tau Yih, Colin Cherry, and Lucy Vanderwende. 2012. MSR SPLAT, a language analysis toolkit. In *Proceedings of the Demonstration Session at the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 21–24, Montréal, Canada, June. Association for Computational Linguistics.

Matthew Richardson, Chris Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Michael Silverstein. 1986. Hierarchy of features and ergativity. In P. Muysken and H. van Riemsdijk, editors, *Features and Projections*, pages 163–232. Foris Publications Holland.

Wen-tau Yih and Vahed Qazvinian. 2012. Measuring word relatedness using heterogeneous vector space

models. In *Proceedings of NAACL-HLT*, pages 616–620, Montréal, Canada, June.

Annie Zaenen, Jean Carletta, Gregory Garretson, Joan Bresnan, Andrew Koontz-Garboden, Tatiana Nikitina, M. Catherine O'Connor, and Tom Wasow. 2004. Animacy encoding in English: Why and how. In Bonnie Webber and Donna K. Byron, editors, *ACL 2004 Workshop on Discourse Annotation*, pages 118–125, Barcelona, Spain, July. Association for Computational Linguistics.