Adding semantic similarity features to Dutch coreference resolution

Wouter de Groot

April 21, 2013

Abstract

In automatic coreference resolution the object is to identify when two noun phrases refer to the same entity in the world. In this paper I use the Dutch language Knack-2002 coreference annotated corpus and example-based supervised machine learning to experiment with adding semantic similarity features to a standard set of linguistic features used in previous work. I use the Cornetto database to add features for WordNet semantic classes and three semantic similarity metrics based on work by Lin (1998), Jiang and Conrath (1997) and Resnik (1995), respectively. Performance is tested using TiMBLs *k*-nearest neighbors algorithm on data split into sets with common noun, proper noun and pronoun anaphors; I find that the metric from Jiang & Conrath improves resolution for all noun types. All features combined improve resolution substantially: 21.1% over baseline for common nouns, 2.7% for pronouns and 11.4% for proper nouns.

1 Introduction

Coreference is the relationship between noun phrases referring to the same entity in the world. Consider an example:

(1) (De interimregering in $(\text{Afghanistan})_1)_2$ maakt bekend dat $(\text{ze})_2$ $(\text{een beroepsleger})_3$ zal oprichten om de vrede en de veiligheid in $(\text{het land})_1$ te garanderen. In $(\text{het leger})_3$ zullen tienduizenden strijders van verschillende Afghaanse krijgsheren worden samengebracht.
<u>*English:*</u> (The interim government in $(\text{Afghanistan})_1)_2$ announces $(\text{it})_2$ will establish $(\text{a professional army})_3$ to guarantee peace and stability in $(\text{the country})_1$. In $(\text{the army})_3$ tens of thousands of warriors from Afghan warlords will be united.

The matching boldface noun phrases in this example corefer. The antecedent *Afghanistan* and anaphor *het land* both refer to a specific country on Earth, for instance. This example illustrates some of the diverse forms of coreference: expressions may be nested, coreference can occur between expressions in different sentences, and three different types of coreferring expressions are recognized: proper nouns like ‘Afghanistan’, common nouns like ‘het land’, and pronouns such as ‘ze’.

Humans resolve coreference all the time, easily interpreting referential expressions in order to keep track of who does what to whom in the discourse. Being able to automatically perform this recognition would help computerised information extraction from human-readable sources like newspapers, books, articles, etc. Additionally, it would allow for better human-machine interaction due to an increased ability to keep track of a conversation. Humans are good judges of semantic similarity (Resnik, 1995) and computational models for human coreference resolution revolve around parsing syntactic information into semantic structures (Wiemer-Hastings and Iacucci, 2001). Based on this I anticipate semantic similarity will contribute to automated coreference resolution: this paper describes an experiment using semantic similarity information to improve automatic coreference resolution for Dutch. Existing work already confirms this usefulness for English (Ponzetto and Strube, 2006), so it will be interesting to see whether these results extend to Dutch.

For this study I will use semantic classes and three different but closely related measures for deriving semantic similarity from a lexical taxonomy as experimental features to improve Dutch coreference resolution. Following Hoste and Daelemans (2004), I will split my data into three to be able to evaluate common nouns, pronouns and proper nouns separately. My expectation is that common nouns benefit from semantic features while pronouns and proper nouns do not: the former type is well represented in lexical taxonomy whereas many proper nouns will not be featured and pronouns will also often corefer with names of people and places.

The remaining sections of this paper are organised as follows: in section 2 I discuss previous work and the machine learning approach used in this study. Section 3 describes the semantic similarity features used while preparation of the data is outlined in section 4. The experimental results are given and evaluated in section 5 and I conclude in section 6 with a discussion.

2 Background

Research into computational coreference resolution models has developed over broadly the past two decades (Ng, 2010). Only a few types of semantic features have been explored so far, like synonymy, antonymy and hypernymy (Bengtson and Roth, 2008) or named entity recognition (Bontcheva, Dimitrov, Maynard, Tablan, and Cunningham, 2002). Ng (2007) uses a variety of semantic features like named entity agreement and a binary semantic similarity judgment according to a similarity-ranked thesaurus which judges pairs of noun phrases semantically similar when either noun phrase has the other as one of its five most similar neighbors. This binary judgment is related to Information Content (introduced fully in section 3.2) in that it is frequency-based: it counts how often noun phrases co-occur to determine how similar they are semantically. He also uses a coreference-annotated corpus to determine the likelihood that a particular noun phrase has an antecedent and encodes this anaphoricity score as a feature; another, similar feature uses probabilities to determine coreferentiality, or the likelihood of two noun phrases corefering. His research obtains encouraging results for semantic similarity: between 2.3 and 7.9 percentage points improvement in F-scores depending on the feature set and the baseline.

For Dutch coreference Hendrickx and Daelemans (2007) used generated clusters of semantically related concepts to bring about a small improvement (about 1%) in resolution F-scores. Hoste and Daelemans (2004) implemented a total of 37 'shallow information sources'–a superset of the features used in Soon, Ng, and Lim (2001)–so called because they are easy to compute using a sentences parser. Using a memory based learning approach with TiMBL (introduced below), they attain an F-score of 51.4 for the set of combined noun types. Their features form the baseline for the current work and are listed in table 1. Five of them are marked with an asterisk to signify that they are not implemented here. Additionally, the CELEX feature, which uses extended gender information like context sensitive genders, is implemented here as gender agreement, as in Soon et al.

Ponzetto and Strube (2006) experimented with the same semantic class and semantic similarity features for English which I implement for Dutch. They group these together as a single set of WordNet (Miller, 1995) features, while they use an additional feature set encoding the semantic roles of the referring expressions and one which takes semantic information from Wikipedia, which organizes its articles into categories to form a kind of taxonomy in itself. They find impressive improvements from adding the WordNet-sourced features (+14.3% accuracy for common nouns on the BNEWS dataset of ACE 2003), slightly smaller results from the Wikipedia-sourced features (+13% for the same dataset), and modest improvements from semantic role features (+4.2% accuracy for pronouns in the merged BNEWS and BWIRE datasets), although there is no distinction between the different noun types there. Ponzetto and Strube conclude that semantic knowledge can help improve coreference resolution over traditional rule-based systems, at least for English.

The Tilburg Memory Based Learner, or TiMBL

DIST_SENT	Number of sentences between anaphor and antecedent
DIST_NP*	Amount of noun phrases between anaphor and antecedent
DIST_LT_THREE	Are there fewer than three sentences between them
local context features	The three words and their parts of speech preceding and following both antecedent and anaphor
I_PRON	Is the antecedent a pronoun
J_PRON	Is the anaphor a pronoun
I+J_PRON	Are both pronouns
J_PRON_I_PROPER	Is the antecedent a proper noun and the anaphor a pronoun
J_DEMON	Is the anaphor demonstrative
J_DEF	Is the anaphor definitive
I_PROPER	In the antecedent a proper noun
J_PROPER	The same for the anaphor
BOTH_PROPER	Are both of them proper nouns
NUM_AGREE	Do the antecedent and anaphor correspond in number
ANA_SYNT	What is the syntactic function of the anaphor (subject, object or predicate)
ANT_SYNT	The same for the antecedent
BOTH_SBJ/OBJ	Do both fulfill the same syntactic function
APPOSITIVE	Is the coreferential noun phrase in apposition to the NP preceding it
COMP_MATCH	Do the strings of anaphor and antecedent fully match
PART_MATCH*	Do they partially match (also uses internal matching, splitting compound words up into their components)
ALIAS	Is either an alias of the other. Used to match IBM to International Business Machines and Homer Simpson to Mr. Simpson
SAME_HEAD*	Do they share the same head (e.g. 'het platteland' and 'het Groningse platteland' have the same head)
CELEX	Gender information for Dutch nouns extended with certain peculiarities like context sensitive genders and female nouns which may be treated as male
SYNONYM*	Are the candidates synonymous
HYPERNYM*	Is either a hypernym of the other
SAME_NE	Are they the same named entity type

Table 1: Table of features, or information sources, as used in Hoste and Daelemans (2004). Items marked with an asterisk are not implemented in the current work. The CELEX feature is implemented as a gender agreement, without the expanded Celex gender information.

(Daelemans, Zavrel, van der Sloot, and van den Bosch, 2009), is a collection of machine learning methods, including k-nearest neighbors (Cover and Hart, 1967) which Hoste and Daelemans use and which is also employed here. K-NN is a classification algorithm which allows classifying new, unseen items based on previously observed training data and works with both numeric and overlap features. Observed training items are stored as vectors in the feature space along with their classifications, and when a new item is presented for classification it is labeled according to a majority of the k items nearest to it in the feature space. For this study the value of k is 1, that is, the classification of the new sample is taken from the single nearest neighbor. This is the default value for TiMBL and in general a fairly safe choice Cover and Hart (1967). The distance weighting (establishing the distance between neighbors) and class voting (calculating the net class of all k neighbors) parameters are also left at their default values: information gain weighting (see section 5 for a short discussion of information gain) and majority voting, respectively. In practice, optimizing these parameters can improve performance in resolving coreference significantly depending on the exact data set used (Hoste, 2005). However, this requires a search over numerous combinations of parameter values–Hoste and Daelemans follow this procedure, too–which falls outside the scope of this study.

3 Adding semantic similarity features

I draw upon two sources of information. The first of these is to look at semantic classes for synsets in the Cornetto (Vossen, Maks, Segers, der Vliet, Moens, Hofmann, Sang, and Rijke, 2013) database and to judge whether the antecedent and anaphor share classes. The second source draws upon frequencies of word occurrences to define the concept of Information Content; similarity here is the amount of information shared between the candidate pair. Together, as discussed above, these are what Ponzetto and Strube (2006) use as their set of WordNet features.

3.1 Semantic class features

Cornetto is a continuation of the Dutch part of EuroWordNet (Vossen, 1998), itself an offshoot of WordNet, and it includes labels from WordNet domains (Magnini and Cavaglià, 2000). These categorize synsets into groups like `astronomy`, `time_period` or `politics`; there are 159 such labels (of which roughly one hundred actually occur in my data). As a small aside about generic concepts: one of these labels is `factotum` which includes synsets of a generic or highly contextualized nature, like colors or numbers–relatively many synsets are classified as a factotum in Cornetto as well as my data. The training data does not consist of synsets, but of words which may correspond to a number of synsets, so semantic class matching is extended to include all synsets corresponding to the words under consideration. The lemma forms (the 'dictionary' form) of the candidate antecedent and anaphor correspond to one or more synsets, so the labels of all of them are extracted, forming two sets of semantic class labels. If these sets of labels for the antecedent and anaphor share at least one member, the semantic classes of the candidate pair are considered matching. In practice, three features record semantic class matching: antecedent label, anaphor label and match. The former two either record the first label found to be matching for both classes, or if no match is found they record the label of the first found synset belonging to the antecedent and anaphor, respectively. The latter is the binary judgment of whether any of the labels match.

To illustrate, consider the antecedent and anaphor pair `partij` ('party') and `kanselier` ('chancellor'). The word `partij` corresponds to six synsets, three of which are labeled `music`, `politics` and `play`, respectively, and three which bear the class label `factotum`. The word `kanselier` corresponds to two synsets, carrying the labels `politics` and `school`. In this example the semantic classes can be matched through `politics`, so the recorded labels for this training instance are `politics` for both antecedent and anaphor, and the match feature is positive. If `partij` had not had a synset with the label `politics` the training instance would have recorded `music` for the antecedent `partij`, `politics` for the anaphor

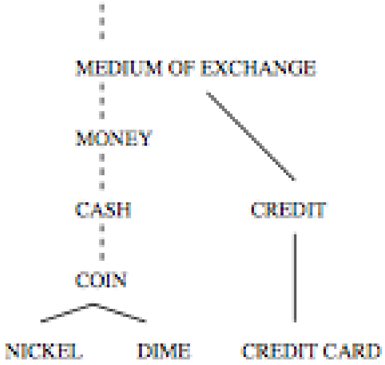

Figure 1: Part of the English WordNet taxonomy. Solid lines represent edges, dotted lines indicate intermediate concepts have been omitted for the example.

`kanselier` and a negative match.

3.2 Information Content features

The other group of features for semantic similarity is based on an approach known as information content. Cornetto's synsets are nodes in a lexical `is-a` taxonomy where each synset or node represents a unique concept. Intuitively, leaf nodes–nodes representing the most specific concepts–convey more information than do nodes further up the taxonomy. The Cornetto taxonomy can be augmented with a word count list to approximate the occurrences of concepts: there is in practise no semantically annotated corpus to give a precise synset frequency count, so one must distribute word occurrences over the appropriate synsets. Using these word counts one can derive relative frequencies, which in turn allow the information content IC of a node in the hierarchy to be given by:

$$IC(c) = -logP(c)$$

where P(c) is the probability of an instance of concept c occurring. IC is thus a range compression of probability. Due to the structure of the taxonomy where nodes subsume the ones below them, the probability increases monotonically as one moves higher up this hierarchy. The information content thus decreases for progressively more abstract concepts. The semantic similarity between two concepts can now be captured by noting the information content of the nearest node which subsume them both.

As an illustration, regard figure 1, taken from Resnik (1995), which shows a subset of the English WordNet. In the left branch of the example *nickel* and *dime* are leaf nodes directly subsumed by *coin*. Their semantic similarity is thus defined by the information content of *coin*. By contrast, *credit card* and *nickel* only find a common subsumer several nodes higher in the hierarchy. Their semantic similarity, therefore, derives from the information content of *medium of exchange*, which will never be greater than that of *coin*.

To evaluate performance I use the information content directly, as in work by Resnik (1995) who compares it to edge counting, human judges and simple probability (i.e. not transforming probability to information content) and finds IC performs significantly better (correlation $r = 0.79$) than edge counting ($r = 0.66$) and probability ($r = 0.67$), although human judges still perform better ($r = 0.90$). This feature, which I call RESNIK, calculates similarity first for synsets:

$$sim(c_1, c_2) = \underset{c \in S(c_1, c_2)}{argmax} [IC(c)]$$

where $S(c_1, c_2)$ are the least common subsumers of concepts c_1 and c_2, and is then extended to words for use with the available training data as follows:

$$sim(w_1, w_2) = \underset{c_1, c_2}{argmax} [sim(c_1, c_2)]$$

In the case of the previous example, the least common subsumer for `partij` and `kanselier` turns out to be `iets` ('something'), which results, unsurprisingly, in a rather low similarity score of 1.22. This is in fact the lowest score possible: the range for RESNIK is 1.22 to ±25, with 0 being used to indicate missing data.

Additionally, I use two derived measures from Jiang and Conrath (1997) and Lin (1998), respectively. Jiang and Conrath integrate information content with edge counting and node weighting and find improvements over Resnik ($r = 0.83$). These hybrid additions are not adopted here, but their formula for deriving similarity from information content in terms of semantic *distance* is adopted in the feature JIANG in the following manner:

$$dist(w_1, w_2) = \underset{c_1,c_2}{argmin}[IC(c_1)+IC(c_2)-2*IC(lcs)]$$

where c_1 and c_2 iterate over all synsets for w_1 and w_2, respectively, and lcs is the lowest common subsumer. The same example as above yield a distance score of 18.51. In this case the scoring range is 0 to ±50 with 10000 representing a missing value. Finally, Lin (1998) redefines the formal definition of semantic similarity in an attempt to provide a better theoretical and universally applicable underpinning. The derivation of similarity is captured in the feature dubbed LIN as follows:

$$sim(w_1, w_2) = \underset{c_1,c_2}{argmax} \frac{2 * IC(lcs)}{IC(c_1) + IC(c_2)}$$

The similarity score for, again, `partij` and `kanselier` according to this measure is 0.12. LIN ranges between 0 and 1 with the lowest possible score, 0, representing missing data as before. In practice, the high (or low, for JIANG) end of these ranges is populated with pairs consisting of identical words: their lowest common subsumer is always the node directly above to them. A fairly high score for different words is found in the instance of `invoering` and `introductie` (both synonyms for ‘introduction’): the score for RESNIK is 13.90 here, 1.70 for JIANG and 0.94 for LIN.

In summary, there are six features encoding semantic similarity. The first two denote the semantic labels in WordNet for the antecedent and anaphor, respectively; feature number three indicates whether these labels are the same. The fourth, fifth and sixth feature represent semantic similarity by using different derivations of information content.

4 Data preparation

For this study the KNACK-2002 corpus (Hoste and Pauw, 2006) was used as the data source, again following Hoste and Daelemans (2004). KNACK-2002 is a coreference annotated corpus compiled from articles of the Flemish magazine Knack which writes about current events, politics, sports, business and similar topics. The corpus contains 267 such articles with a total of 104736 words, ranging from 33 to 2640 words per article with a median of 167 words. Antecedents and anaphors in these articles are tagged; there are 12579 coreference tags in the corpus (median 19 per article), together they form 5329 chains of coreference (median 8 per article). Due to a processing error 25 articles were not used*.

A training instance is a single line based on a specific antecedent and anaphor candidate pair and contains a comma-separated line with all of the the semantic similarity features described previously, the baseline features and a positive or negative classification: the candidate pair does, or does not, corefer. Final training instances look like this (using the same example of `partij` and `kanselier`):

> politics, politics, +, 0.116655821287, 18.5084620419, 1.22212830077, [...], negative

To construct these instances a coreference tag is matched with another one in the same sentence or at most 20 sentences prior, as in Hoste and Daelemans. This allows for two anaphoric tags to be matched, two antecedents, or even a tag to itself. The first of the two selected tags now serves as antecedent, the other as anaphor. Frog, a morpho-syntactic sentence parser (Van den Bosch, Busser, Daelemans, and Canisius, 2007) is first used to gather the baseline information features. The semantic class label and Information Content features are extracted from the augmented Cornetto database using the lemmas of the antecedent and anaphor tags, and are prepended to the instance. Because I want to test performance on common nouns, pronouns and proper nouns separately the data are split according to the noun type of the anaphor; examples with anaphors which are not nouns at all are discarded. The total and per-type amount of training data can be found in table 2. I compensate for the imbalance between

*specifically: 01BWEEKQ_216, 02BWEEKQ_206, 02WWEEKQ_224, 04WWEEKQ_227, 06ONTDAR_257, 07ONTDK1_DDR, 08BWEEKQ_262, 10ONTDAR_204, VANDER00_RVC, 02BWEEKQ_200, 02ONTDK1_218, 03ONTDK1_180, 05WETSTR_FRO, 06WWEEKQ_259, 07ONTDK2_DDR3 09WOLFOW_HVH, DATPROBL_GDS, 02BWEEKQ_201, 02WWEEKQ_222, 04ONTDK1_196, 06BWEEKQ_238, 07BWEEKQ_265, 08BWEEKQ_261, 09WWEEKQ_265, EURO1111_RVC

set	positive	negative	total	% positive
common	3997	151115	155112	2.58
pronoun	2659	86418	89077	2.99
proper	2882	108372	111254	2.59
total	9538	345905	355443	2.68

Table 2: Amount of training examples per noun type, their totals and proportions.

positive and negative instances by evaluating performance only on the classifications of positive examples. This precaution is necessary because the overwhelming amount of negative instances means that simply guessing a classification as negative is correct a lot of the time, which would unrealistically drive up scores. Alternative methods for dealing with imbalanced data sets are discussed by Hoste (2005), like down-sampling the majority class or adjusting the cost of misclassification.

This task of adding semantic similarity information suffers from sparse word count data in Cornetto. For common nouns 54% of training instances are unable to use semantic similarity. Pronouns miss semantic similarity in 71% of cases and for proper nouns data is unavailable in 93% of all training instances. Semantic class labels suffer from a similar problem: not all of the synsets in Cornetto have been annotated with WordNet domains. The numbers are nearly identical to the missing similarity data in terms of missing at least one label: 54% of instances for common nouns, 71% for pronouns and 95% for proper nouns.

5 Evaluation

Leave-one-out cross-validation can be regarded as a specialised version of k-fold cross-validation. In k-fold cross-validation one partitions the data in k equal parts and performs training k times on all data except one of the segments which serves as training. For leave-one-out k equals n, the amount of training instances. Thus, for n repetitions one of the training instances is set aside while training takes place on all the other data, after which the lone datum is classified. Leave-one-out ensures testing always happens on unseen data, like k-fold cross-validation, while maximising the amount of data available for training.

A total of six cases were tested with TiMBL's 1-nearest neighbor algorithm in a leave-one-out cross-validation configuration: baseline only, baseline with added semantic class labels, with LIN, with JIANG, with RESNIK, and with all features combined. Table 3 shows the F-scores for the positive training instances obtained by the classifier for these six conditions.

McNemar's chi-squared test (Everitt, 1977) is used to ascertain the statistical significance of any performance differences. In it, the performance data of two classifiers are arranged in a contingency table as illustrated in table 4a. Of interest are cells `b` and `c` which represent the difference between the two–as an aside, he counts in cells `a` and `d` could be used as a rough indicator of the difficulty of the problem: they indicate either that both classifiers struggle with a lot of the data (cell `a` is overrepresented) or that they find many of the data easy to classify (cell `d` stands out). The expectation under the null hypothesis is that there is no difference in classifiers, and that therefore `b` = `c` as illustrated in table 4b. The test produces a χ^2 statistic as follows:

$$\chi^2 = \frac{(|b-c|-1)^2}{b+c}$$

The contingency tables for McNemar's test are always 2x2 which means one degree of freedom, so the critical value is always $\chi^2 = 3.8415$ at the conventional $\alpha = 0.05$ level.

The results of McNemar analyses for each feature compared against baseline separated per noun type are arranged in table 5; significant values mean the feature performs better than baseline. Common noun resolution is improved by each of the features, as expected, with a 14.5% gain for RESNIK. Pronouns benefit from the

Set	baseline	class	LIN	JIANG	RESNIK	Combined
common	0.275	0.288	0.292	0.286	0.315	0.333
pronoun	0.228	0.248	0.230	0.232	0.232	0.254
proper	0.403	0.408	0.412	0.411	0.416	0.414

Table 3: F-scores for the different conditions; positive training instances, only.

a: Misclassified by A and B	b: Correctly classified by A
c: Correctly classified by B	d: Correctly classified by A and B

(a) Table structure

a	(b + c) / 2
(b + c) / 2	d

(b) Null hypothesis

Table 4: Contingency table and null hypothesis for McNemar's test.

addition of semantic classes but not much from any of the Information Content features: resolution improves 8.8% with semantic class labels and 1.8% with JIANG. Proper nouns, lastly, display the opposite picture to pronouns: semantic classes do not help here, but information content does with RESNIK improving resolution 3.2%. Looking at the results from a per-feature angle, it appears that JIANG is the single most informative feature with significant results at the conventional level for each type of nouns. LIN and RESNIK both do well on common nouns and proper nouns but fail to contribute to pronoun resolution while semantic classes, conversely, do improve pronouns as well as common nouns but leave proper noun resolution no better off.

For JIANG there are fewer default ('missing information') values entered than for either RESNIK or LIN which have roughly equal amounts of missing data. This observation is puzzling: since all three equations use the same information either all three measures should work for a particular instance, or otherwise none should. This suggests an unforeseen programming mistake in the pycornetto[†] library which I use to compute the measures from Cornetto. Of course, the way it extracts distance from the Information Content of the synsets under evaluation and their lowest common subsumer could simply be superior.

The results for proper nouns and pronouns are somewhat unexpected. For pronouns it appears there are relatively many antecedents which are professions which, when coupled with a pronoun, yield a modest similarity score above the minimum. For example, the pair `handelaar` ('trader') and `hij` ('he') scores 0.43 for LIN, 11,74 for JIANG and 4.37 for RESNIK. There are also many cases where similarity information is not available because the antecedent is a person whose name does not occur in the taxonomy, so perhaps the combination of a pronoun anaphor and missing similarity information is in itself informative. In the case of proper nouns, the Knack corpus references major cities relatively often due to its political news coverage, and names of places like Paris and Moscow are in Cornetto and yield high similarity scores when paired to another city, or the country they are in. On the other hand, many proper noun antecedents are names of politicians for which Cornetto offers no information.

An analysis of the combined feature set is listed in table 6. Performance of the set of all features combined is tested against baseline and each separate feature in turn; significant values in this table indicate the combination performs better. By performing better than baseline for all three noun types (21.1% for common nouns, 11.4% for pronouns and 2.7% for proper nouns), the combination is at least as good as JIANG, the best single feature. It also, however, outperforms all separate features for common nouns and pronouns. For proper nouns there is no improvement except when compared to semantic classes. In sum, combining both

[†] http://code.google.com/p/pycornetto/

the semantic class and all three information content features yields performance at least as good as the best performing single feature, and outclasses each separate one for common nouns and pronouns; it does not outperform proper nouns, however, compared to any of the information content features.

Lastly, a look at the information gain ratio for the features in this study. Information gain is a per-feature measure indicating essentially the difference in uncertainty about a training instance's classification with or without that feature. TiMBL produces a permutation of features based on their relative normalized information gain: their gain ratio. One intuitively expects highly contributive information sources to feature prominently in these permutations. The semantic class label match feature scores well in this ranking: for proper nouns it is considered the most informative, while it is 4^{th} for common nouns and 9^{th} for proper nouns, out of a possible 39. The discrepancy between permutation ranking and contribution to performance, especially in the case of proper nouns, is telling about the predictive power of the information gain ratio. The remaining features do not score prominently: RESNIK hovers around position 11, JIANG and LIN are near the bottom and the semantic class labels themselves can be found in the middle. Even though the correlation between information gain and performance is less than perfect the permutation of features does corroborate the experimental results: semantic similarity features are contributive but do not carry the performance single-handedly.

6 Discussion

When compared to Hoste and Daelemans (2004), one thing immediately stands out: the baseline scores in the present study are significantly lower, even though I use the same corpus and nearly the same feature set. It is likely the missing features account for at least part of this gap: PART_MATCH, the unimplemented feature indicating whether the two candidate strings partially match, performs internal matching and Dutch does have many cases where compound words may be split for coreference (e.g. `tribunaal` serving as an anaphor for `oorlogstribunaal`–'tribunal' and 'war tribunal'); Soon et al. (2001) do find that string matching features are the most contributive overall. It is also likely the SYNONYM and HYPERNYM features would have contributed to performance. Hoste and Daelemans' use of the optimization approach in Hoste (2005) will have boosted their performance scores, too, so it is difficult to establish exactly which modification produces (most of) the performance disparity. If baseline performance were better, it is possible the effect from adding the experimental features might shrink or disappear. However, as discussed previously, Ponzetto and Strube (2006) also found an improvement with these features, for English. It seems plausible, therefore, that semantics are robust and that this effect is not the result of an impoverished baseline.

The reason why JIANG outperforms RESNIK and LIN might be, as discussed above, due to a programming error. Especially RESNIK is nearly significant for pronouns at the conventional level, which puts it almost on par with JIANG. RESNIK uses the IC of the most informative lowest common subsumer directly while the JIANG and LIN measures clearly still favor the highest possible lowest common subsumer: they all essentially express the same information in slightly different ways. Under the hypothesis that JIANG indeed performs more favorably due to an error, the conclusion could well be that there is no appreciable difference between the different similarity measures evaluated here. A more detailed analysis could provide conclusive answers in this regard.

Further work could focus on refining the selection of baseline and experimental features to minimize redundancy and arrive at an optimal combination of semantic information instead of simply combining them all. Improving the baseline to more closely replicate the results from previous work would also shed light on the robustness of the informativeness of the semantic similarity features evaluated here; using genetic algorithms to optimize the memory based learning algorithm's parameters for the specific task at hand is a likely road to success. The integration of Wikipedia's taxonomic properties is an encouraging addition for English coreference resolution; perhaps similar

Set	Semantic class			LIN		
	χ^2	p	Sig.	χ^2	p	Sig.
common	9.54	0.002	**	14.77	<0.001	***
pronoun	17.23	<0.001	***	1.96	0.161	
proper	3.43	0.064	.	8.26	0.004	**
	JIANG			RESNIK		
	χ^2	p	Sig.	χ^2	p	Sig.
common	15.57	<0.001	***	48.39	<0.001	***
pronoun	4.63	0.031	*	3.35	0.067	.
proper	8.58	0.003	**	12.18	<0.001	***

Table 5: Comparison of the different features against baseline using McNemar's test. Significant values of *p* indicate the feature performs better than baseline. The contingency table for each test is 2x2, so there is always 1 degree of freedom.

Set	Baseline			Semantic class			LIN		
	χ^2	p	Sig.	χ^2	p	Sig.	χ^2	p	Sig.
common	89.49	<0.001	***	69.30	<0.001	***	51.66	<0.001	***
pronoun	26.80	<0.001	***	7.20	0.007	**	22.67	<0.001	***
proper	9.84	0.002	**	5.09	0.024	*	1.48	0.224	
	JIANG			RESNIK					
	χ^2	p	Sig.	χ^2	p	Sig.			
common	55.42	<0.001	***	15.94	<0.001	***			
pronoun	18.79	<0.001	***	20.92	<0.001	***			
proper	1.28	0.258		0.15	0.700				

Table 6: Comparison of set with all features combined against separate features. Significant values of *p* indicate the combined feature set performs better. Again all tests have a single degree of freedom.

improvements may be obtained for Dutch, too. Having better semantic information in Cornetto (word counts, semantic class labels), lastly, would most likely serve to add robustness to the results as well.

Solving coreferring expressions is one of the major challenges in natural language processing. Ongoing research in this area will aid Artificial Intelligence systems wherever natural language must be dealt with, like parsing messages or interacting with people in a comfortable, natural way. Semantics provide a promising path to future improvements in this area and a biologically plausible one at that, but as my results show, there is a long way to go yet before automatic coreference resolution matches the judgment of humans.

In this paper I have presented a semantic similarity approach to coreference resolution in Dutch using a memory based machine learning approach. As predicted, common noun resolution benefits from semantics, yielding a 21.1% better performance when a combination of all of the information content features together with the semantic class labels is used. Unexpectedly, however, smaller but still statistically significant improvements are found with this set for the pronoun (2.7%) and proper noun (11.4%) types. Finally, the JIANG feature is the best performing single feature and the combined set never performs worse than any single feature.

Acknowledgements

I am grateful to Erwin Marsi for supplying me with a copy of the word count list compiled by Antal van den Bosch and Walter Daelemans to augment Cornetto with. I am furthermore grateful to Johan Los for dealing with much of the preprocessing of extracting the training pairs and baseline features from KNACK-2002.

References

E. Bengtson and D. Roth. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics, 2008.

K. Bontcheva, M. Dimitrov, D. Maynard, V. Tablan, and H. Cunningham. Shallow methods for named entity coreference resolution. In *Chaînes de références et résolveurs danaphores, workshop TALN*, 2002.

T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. *TiMBL: Tilburg Memory Based Learner, version 6.3, Reference Guide*, 2009. Available from http://ilk.uvt.nl/downloads/pub/papers/ilk1001.pdf.

B.S. Everitt. *The analysis of contingency tables.* Chapman and Hall, London, 1977.

I. Hendrickx and W. Daelemans. Adding semantic information: unsupervised clusters for coreference resolution, 2007.

V. Hoste. *Optimization issues in machine learning of coreference resolution.* PhD thesis, University of Antwerp, 2005.

V. Hoste and W. Daelemans. Learning dutch coreference resolution. In *Proceedings of the 15th Computational Linguistics in Netherlands Meeting*, pages 133–148, 2004.

V. Hoste and G. De Pauw. Knack-2002: a richly annotated corpus of dutch written text. In *The fifth international conference on Language Resources and Evaluation.* European Language Resources Association, 2006.

J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research on Computational Linguistics*, 1997.

D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th international conference on Machine Learning*, pages 296–304, 1998.

B. Magnini and G. Cavaglià. Integrating subject field codes into wordnet. In *Proceedings of the Second International Conference Language Resources and Evaluation Conference*, pages 1413–1418, 2000.

G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

V. Ng. Shallow semantics for coreference resolution. In *Proceedings of IJCAI*, pages 1689–1694, 2007.

V. Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411. Association for Computational Linguistics, 2010.

S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. pages 192–199. Association for Computational Linguistics, 2006.

P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 448–453. Morgan Kaufmann, 1995.

W. Soon, H. Ng, and D. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

A. Van den Bosch, G. J. Busser, W. Daelemans, and S. Canisius. An efficient memory-based morphosyntactic tagger and parser for dutch. In *Computational linguistics in the Netherlands: Selected papers from the Seventeenth CLIN Meeting*, pages 99–114, 2007.

P. Vossen. *EuroWordNet: a multilingual database with lexical semantic networks.* Kluwer Academic, Boston, 1998.

P. Vossen, I. Maks, R. Segers, H. Van der Vliet, M-F. Moens, K. Hofmann, E. Tjong Kim Sang, and M. De Rijke. Cornetto: a combinatorial lexical semantic database for dutch. In P. Spyns and J. Odijk, editors, *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing, chapter 10. Springer, 2013.

P. Wiemer-Hastings and C. Iacucci. A computational model of human coreference judgements. In M. Poesio, editor, *Proceedings of the First Workshop on Cognitively Plausible Models of Semantic Processing (SEMPRO 2001). Workshop held in conjunction with the Annual Meeting of the Cognitive Science Society.* University of Edinburgh, Human Communication Research Centre, 2001.