Glen, Glenda or Glendale: Unsupervised and Semi-supervised Learning of English Noun Gender

Shane Bergsma
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada, T6G 2E8
bergsma@cs.ualberta.ca

Dekang Lin
Google, Inc.
1600 Amphitheatre Parkway
Mountain View
California, 94301
lindek@google.com

Randy Goebel
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada, T6G 2E8
goebel@cs.ualberta.ca

Abstract

English pronouns like *he* and *they* reliably reflect the gender and number of the entities to which they refer. Pronoun resolution systems can use this fact to filter noun candidates that do not agree with the pronoun gender. Indeed, broad-coverage models of noun gender have proved to be the most important source of world knowledge in automatic pronoun resolution systems.

Previous approaches predict gender by counting the co-occurrence of nouns with pronouns of each gender class. While this provides useful statistics for frequent nouns, many infrequent nouns cannot be classified using this method. Rather than using co-occurrence information directly, we use it to automatically annotate training examples for a large-scale discriminative gender model. Our model collectively classifies all occurrences of a noun in a document using a wide variety of contextual, morphological, and categorical gender features. By leveraging large volumes of unlabeled data, our full semi-supervised system reduces error by 50% over the existing state-of-the-art in gender classification.

1 Introduction

Pronoun resolution is the process of determining which preceding nouns are referred to by a particular pronoun in text. Consider the sentence:

(1) Glen told Glenda that she was wrong about Glendale.

A pronoun resolution system should determine that the pronoun *she* refers to the noun *Glenda*. Pronoun resolution is challenging because it requires a lot of *world knowledge* (general knowledge of word types). If *she* is replaced with the pronoun *he* in (1), *Glen* becomes the antecedent. Pronoun resolution systems need the knowledge of *noun gender* that advises that *Glen* is usually masculine (and thus referred to by *he*) while *Glenda* is feminine.

English third-person pronouns are grouped in four gender/number categories: masculine (*he, his, him, himself*), feminine (*she, her, herself*), neutral (*it, its, itself*), and plural (*they, their, them, themselves*). We broadly refer to these gender and number classes simply as *gender*. The objective of our work is to correctly assign gender to English noun tokens, in context; to determine which class of pronoun will refer to a given noun.

One successful approach to this problem is to build a statistical gender model from a noun's association with pronouns in text. For example, Ge et al. (1998) learn *Ford* has a 94% chance of being neutral, based on its frequent co-occurrence with neutral pronouns in text. Such estimates are noisy but useful. Both Ge et al. (1998) and Bergsma and Lin (2006) show that learned gender is the most important feature in their pronoun resolution systems.

English differs from other languages like French and German in that gender is not an inherent grammatical property of an English noun, but rather a property of a real-world entity that is being referred to. A common noun like *lawyer* can be (semantically) masculine in one document and feminine in another. While previous statistical gender models learn gender for noun types only, we use document context to correctly determine the current gender class of noun tokens, making dynamic decisions on common nouns like *lawyer* and ambiguous names like *Ford*. Furthermore, if a noun type has not yet

Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), pages 120–128,
Boulder, Colorado, June 2009.

been observed (an unknown word), previous approaches cannot estimate the gender. Our system, on the other hand, is able to correctly determine that unknown words *corroborators* and *propeller-heads* are plural, while *Pope Formosus* is masculine, using learned contextual and morphological cues.

Our approach is based on the key observation that while gender information from noun-pronoun co-occurrence provides imperfect noun coverage, it can nevertheless provide rich and accurate training data for a large-scale discriminative classifier. The classifier leverages a wide variety of noun properties to *generalize* from the automatically-labeled examples. The steps in our approach are:

1. Training:

(a) Automatically extract a set of seed *(noun,gender)* pairs from high-quality instances in a statistical gender database.
(b) In a large corpus of text, find documents containing these nouns.
(c) For all instances of each noun in each document, create a single, composite feature vector representing all the contexts of the noun in the document, as well as encoding other selected properties of the noun type.
(d) Label each feature vector with the seed noun's corresponding gender.
(e) Train a 4-way gender classifier (masculine, feminine, neutral, plural) from the automatically-labeled vectors.

2. Testing:

(a) Given a new document, create a composite feature vector for all occurrences of each noun.
(b) Use the learned classifier to assign gender to each feature vector, and thus all occurrences of all nouns in the document.

This algorithm achieves significantly better performance than the existing state-of-the-art statistical gender classifier, while requiring no manually-labeled examples to train. Furthermore, by training on a small number of manually-labeled examples, we can combine the predictions of this system with the counts from the original gender database. This semi-supervised extension achieves 95.5% accuracy on final unseen test data, an impressive 50% reduction in error over previous work.

2 Path-based Statistical Noun Gender

Seed *(noun,gender)* examples can be extracted reliably and automatically from raw text, providing the training data for our discriminative classifier. We call these examples *pseudo*-seeds because they are created fully automatically, unlike the small set of manually-created seeds used to initialize other bootstrapping approaches (cf. the bootstrapping approaches discussed in Section 6).

We adopt a statistical approach to acquire the pseudo-seed *(noun,gender)* pairs. All previous statistical approaches rely on a similar observation: if a noun like *Glen* is often referred to by masculine pronouns, like *he* or *his*, then *Glen* is likely a masculine noun. But for most nouns we have no annotated data recording their coreference with pronouns, and thus no data from which we can extract the co-occurrence statistics. Thus previous approaches rely on either hand-crafted coreference-indicating patterns (Bergsma, 2005), or iteratively guess and improve gender models through expectation maximization of pronoun resolution (Cherry and Bergsma, 2005; Charniak and Elsner, 2009). In statistical approaches, the more frequent the noun, the more accurate the assignment of gender.

We use the approach of Bergsma and Lin (2006), both because it achieves state-of-the-art gender classification performance, and because a database of the obtained noun genders is available online.[1] Bergsma and Lin (2006) use an unsupervised algorithm to identify syntactic paths along which a noun and pronoun are highly likely to corefer. To extract gender information, they processed a large corpus of news text, and obtained co-occurrence counts for nouns and pronouns connected with these paths in the corpus. In their database, each noun is listed with its corresponding masculine, feminine, neutral, and plural pronoun co-occurrence counts, e.g.:

glen	555	42	32	34
glenda	8	102	0	11
glendale	24	2	167	18
glendalians	0	0	0	1
glenn	3182	207	95	54
glenna	0	6	0	0

[1] Available at http://www.cs.ualberta.ca/~bergsma/Gender/

This sample of the gender data shows that the noun *glenda*, for example, occurs 8 times with masculine pronouns, 102 times with feminine pronouns, 0 times with neutral pronouns, and 11 times with plural pronouns; 84% of the time *glenda* co-occurs with a feminine pronoun. Note that all nouns in the data have been converted to lower-case.[2]

There are gender counts for 3.1 million English nouns in the online database. These counts form the basis for the state-of-the-art gender classifier. We can either take the most-frequent pronoun-gender (MFPG) as the class (e.g. $feminine$ for *glenda*), or we can supply the logarithm of the counts as features in a 4-way multi-class classifier. We implement the latter approach as a comparison system and refer to it as PATHGENDER in our experiments.

In our approach, rather than using these counts directly, we process the database to automatically extract a high-coverage but also high-quality set of pseudo-seed *(noun,gender)* pairs. First, we filter nouns that occur less than fifty times and whose MFPG accounts for less than 85% of counts. Next, we note that the most reliable nouns should occur relatively often in a coreferent path. For example, note that *importance* occurs twice as often on the web as *Clinton*, but has twenty-four times less counts in the gender database. This is because *importance* is unlikely to be a pronoun's antecedent. We plan to investigate this idea further in future work as a possible filter on antecedent candidates for pronoun resolution. For the present work, simply note that a high ratio of database-count to web-count provides a good indication of the reliability of a noun's gender counts, and thus we filter nouns that have such ratios below a threshold.[3] After this filtering, we have about 45 thousand nouns to which we automatically assign gender according to their MFPG. These *(noun,gender)* pairs provide the seed examples for the training process described in the following section. Figure 1 provides a portion of the ordered $feminine$ seed nouns that we extracted.

[2]Statistical approaches can adapt to the idiosyncrasies of the particular text domain. In the news text from which this data was generated, for example, both the word *ships* and specific instances of ships (*the USS Cole*, *the Titanic*, etc.) are neutral. In Wikipedia, on the other hand, feminine pronouns are often used for ships. Such differences can be learned automatically.

[3]We roughly tuned all the thresholds to obtain the highest number of seeds such that almost all of them looked correct (e.g. Figure 1). Further work is needed to determine whether a different precision/recall tradeoff can improve performance.

```
...
stefanie
steffi graf
steinem
stella mccartney
stellar jayne
stepdaughter
stephanie
stephanie herseth
stephanie white
stepmother
stewardess
...
```

Figure 1: Sample $feminine$ seed nouns

3 Discriminative Learning of Gender

Once we have extracted a number of pseudo-seed *(noun,gender)* pairs, we use them to automatically-label nouns (in context) in raw text. The auto-labeled examples provide training data for discriminative learning of noun gender.

Since the training pairs are acquired from a sparse and imperfect model of gender, what can we gain by training over them? We can regard the Bergsma and Lin (2006) approach and our discriminative system as two orthogonal views of gender, in a co-training sense (Blum and Mitchell, 1998). Some nouns can be accurately labeled by noun-pronoun co-occurrence (a view based on pronoun co-occurrence), and these examples can be used to deduce other gender-indicating regularities (a view based on other features, described below).

We presently explain how examples are extracted using our pseudo-seed pairs, turned into auto-labeled feature vectors, and then used to train a supervised classifier.

3.1 Automatic example extraction

Our example-extraction module processes a large collection of documents (roughly a million documents in our experiments). For each document, we extract all the nouns, including context words within ± 5 tokens of each noun. We then group the nouns by

Class=$masculine$	*String*="Lee"
Contexts =	
"led some to suggest that $\star$, who was born in"	
"$\star$ also downloaded secret files to"	
"$\star$ says he was just making"	
"by mishandling the investigation of $\star$."	
...	

Figure 2: Sample noun training instance

their (lower-case) string. If a group's noun-string is in our set of seed *(noun,gender)* pairs, we assign the corresponding *gender* to be the class of the group. Otherwise, we discard the group. To prevent frequent nouns from dominating our training data, we only keep the first 200 groups corresponding to each noun string. Figure 2 gives an example training noun group with some (selected) context sentences. At test time, all nouns in the test documents are converted to this format for further processing.

We group nouns because there is a strong tendency for nouns to have only one sense (and hence gender) per discourse. We extract contexts because nearby words provide good clues about which gender is being used. The notion that nouns have only one sense per discourse/collocation was also exploited by Yarowsky (1995) in his seminal work on bootstrapping for word sense disambiguation.

3.2 Feature vectors

Once the training instances are extracted, they are converted to labeled feature vectors for supervised learning. The automatically-determined gender provides the class label (e.g., $masculine$ for the group in Figure 2). The features identify properties of the noun and its context that potentially correlate with a particular gender category. We divide the features into two sets: those that depend on the contexts within the document (Context features: features of the *tokens* in the document), and those that depend on the noun string only (Type features). In both cases we induce the feature space from the training examples, keeping only those features that occur more than 5 times.

3.2.1 Context features

The first set of features represent the contexts of the word, using all the contexts in the noun group. To illustrate the potential utility of the context information, consider the context sentences for the masculine noun in Figure 2. Even if these snippets were all the information we were given, it would be easy to guess the gender of the noun.

We use binary attribute-value features to flag, for any of the contexts, the presence of all words at context positions $\pm 1, \pm 2$, etc. (sometimes called *collocation* features (Golding and Roth, 1999)). For example, feature 255920 flags that the word two-to-the-right of the noun is *he*. We also provide features for the presence of all words *anywhere* within ± 5 tokens of the noun (sometimes called *context words*). We also parse the sentence and provide a feature for the noun's parent (and relationship with the parent) in the parse tree. For example, the instance in Figure 2 has features *downloaded(subject)*, *says(subject)*, etc. Since plural nouns should be governed by plural verbs, this feature is likely to be especially helpful for number classification.

3.2.2 Type features

The next group of features represent morphological properties of the noun. Binary features flag the presence of all prefixes and suffixes of one-to-four characters. For multi-token nouns, we have features for the first and last token in the noun. Thus we hope to learn that *Bob* begins masculine nouns while *inc.* ends neutral ones.

Finally, we have features that indicate if the noun or parts of the noun occur on various lists. Indicator features specify if any token occurs on in-house lists of given names, family names, cities, provinces, countries, corporations, languages, etc. A feature also indicates if a token is a corporate designation (like *inc.* or *ltd.*) or a human one (like *Mr.* or *Sheik*). We also made use of the person-name/instance pairs automatically extracted by Fleischman et al. (2003).[4] This data provides counts for pairs such as (Zhang Qiyue, *spokeswoman*) and (Thorvald Stoltenberg, *mediator*). We have features for all *concepts* (like *spokeswoman* and *mediator*) and therefore learn their association with each gender.

3.3 Supervised learning and classification

Once all the feature vectors have been extracted, they are passed to a supervised machine learn-

[4] Available at http://www.mit.edu/~mbf/instances.txt.gz

ing algorithm. We train and classify using a multi-class linear-kernel Support Vector Machine (SVM) (Crammer and Singer, 2001). SVMs are maximum-margin classifiers that achieve good performance on a range of tasks. At test time, nouns in test documents are processed exactly as the training instances described above, converting them to feature vectors. The test vectors are classified by the SVM, providing gender classes for all the nouns in the test document. Since all training examples are labeled automatically (auto-trained), we denote systems using this approach as -AUTO.

3.4 Semi-supervised extension

Although a good gender classifier can be learned from the automatically-labeled examples alone, we can also use a small quantity of gold-standard labeled examples to achieve better performance.

Combining information from our two sets of labeled data is akin to a domain adaptation problem. The gold-standard data can be regarded as high-quality in-domain data, and the automatically-labeled examples can be regarded as the weaker, but larger, out-of-domain evidence.

There is a simple but effective method for combining information from two domains using predictions as features. We train a classifier on the full set of automatically-labeled data (as described in Section 3.3), and then use this classifier's predictions as features in a separate classifier, which is trained on the gold-standard data. This is like the competitive *Feats* domain-adaptation system in Daumé III and Marcu (2006).

For our particular SVM classifier (Section 4.1), predictions take the form of four numerical scores corresponding to the four different genders. Our gold-standard classifier has features for these four predictions plus features for the original path-based gender counts (Section 2).[5] Since this approach uses both automatically-labeled and gold-standard data in a semi-supervised learning framework, we denote systems using this approach as -SEMI.

[5]We actually use 12 features for the path-based counts: the 4 original, and then 4 each for counts for the first and last token in the noun string. See PATHGENDER+ in Section 4.2.

4 Experiments

4.1 Set-up

We parsed the 3 GB AQUAINT corpus (Vorhees, 2002) using Minipar (Lin, 1998) to create our unlabeled data. We process this data as described in Section 3, making feature vectors from the first 4 million noun groups. We train from these examples using a linear-kernel SVM via the the efficient SVM^{multiclass} instance of the SVM^{struct} software package (Tsochantaridis et al., 2004).

To create our gold-standard gender data, we follow Bergsma (2005) in extracting gender information from the anaphora-annotated portion[6] of the American National Corpus (ANC) (Ide and Suderman, 2004). In each document, we first group all nouns with a common lower-case string (exactly as done for our example extraction (Section 3.1)). Next, for each group we determine if a third-person pronoun refers to any noun in that group. If so, we label all nouns in the group with the gender of the referring pronoun. For example, if the pronoun *he* refers to a noun *Brown*, then all instances of *Brown* in the document are labeled as masculine. We extract the genders for 2794 nouns in the ANC training set (in 798 noun groups) and 2596 nouns in the ANC test set (in 642 groups). We apply this method to other annotated corpora (including MUC corpora) to create a development set.

The gold standard ANC training set is used to set the weights on the counts in the PATHGENDER classifiers, and to train the semi-supervised approaches. We also use an SVM to learn these weights. We use the development set to tune the SVM's regularization parameter, both for systems trained on automatically-generated data, and for systems trained on gold-standard data. We also optimize each automatically-trained system on the development set when we include this system's predictions as features in the semi-supervised extension. We evaluate and state performance for all approaches on the final unseen ANC test set.

4.2 Evaluation

The primary purpose of our experiments is to determine if we can improve on the existing state-of-the-art in gender classification (path-based gender

[6]Available at http://www.cs.ualberta.ca/~bergsma/CorefTags/

counts). We test systems both trained purely on automatically-labeled data (Section 3.3), and those that leverage some gold-standard annotations in a semi-supervised setting (Section 3.4). Another purpose of our experiments is to investigate the relative value of our context-based features and type-based features. We accomplish these objectives by implementing and evaluating the following systems:

1. **PATHGENDER**:
   A classifier with the four path-based gender counts as features (Section 2).
2. **PATHGENDER+**:
   A method of back-off to help classify unseen nouns: For multi-token nouns (like *Bob Johnson*), we also include the four gender counts aggregated over all nouns sharing the first token (*Bob* .*), and the four gender counts over all nouns sharing the last token (.* *Johnson*).
3. **CONTEXT-AUTO**:
   Auto-trained system using only context features (Section 3.2.1).
4. **TYPE-AUTO**:
   Auto-trained system using only type features (Section 3.2.2).
5. **FULL-AUTO**:
   Auto-trained system using all features.
6. **CONTEXT-SEMI**:
   Semi-sup. combination of the PATHGENDER+ features and the CONTEXT-AUTO predictions.
7. **TYPE-SEMI**:
   Semi-sup. combination of the PATHGENDER+ features and the TYPE-AUTO predictions.
8. **FULL-SEMI**:
   Semi-sup. combination of the PATHGENDER+ features and the FULL-AUTO predictions.

We evaluate using *accuracy*: the percentage of labeled nouns that are correctly assigned a gender class. As a baseline, note that always choosing *neutral* achieves 38.1% accuracy on our test data.

5 Results and Discussion

5.1 Main results

Table 1 provides our experimental results. The original gender counts already do an excellent job classifying the nouns; PATHGENDER achieves 91.0% accuracy by looking for exact noun matches. Our simple method of using back-off counts for the first and last token, PATHGENDER+, achieves 92.1%. While PATHGENDER+ uses gold standard data to determine optimum weights on the twelve counts, FULL-AUTO achieves 92.6% accuracy using no gold standard training data. This confirms that our algorithm, using no manually-labeled training data, can produce a competitive gender classifier.

1. PATHGENDER	91.0
2. PATHGENDER+	92.1
3. CONTEXT-AUTO	79.1
4. TYPE-AUTO	89.1
5. FULL-AUTO	92.6
6. CONTEXT-SEMI	92.4
7. TYPE-SEMI	91.3
8. FULL-SEMI	**95.5**

Table 1: Noun gender classification accuracy (%)

Both PATHGENDER and PATHGENDER+ do poorly on the noun types that have low counts in the gender database, achieving only 63% and 66% on nouns with less than ten counts. On these same nouns, FULL-AUTO achieves 88% performance, demonstrating the robustness of the learned classifier on the most difficult examples for previous approaches (FULL-SEMI achieves 94% on these nouns).

If we break down the contribution of the two feature types in FULL-AUTO, we find that we achieve 89.1% accuracy by only using type features, while we achieve 79.1% with only context features. While not as high as the type-based accuracy, it is impressive that almost four out of five nouns can be classified correctly based purely on the document context, using no information about the noun itself. This is information that has not previously been systematically exploited in gender classification models.

We examine the relationship between training data size and accuracy by plotting a (logarithmic-scale) learning curve for FULL-AUTO (Figure 3). Although using four million noun groups originally seemed sufficient, performance appears to still be increasing. Since more training data can be generated automatically, it appears we have not yet reached the full power of the FULL-AUTO system. Of course, even with orders of magnitude more data, the system

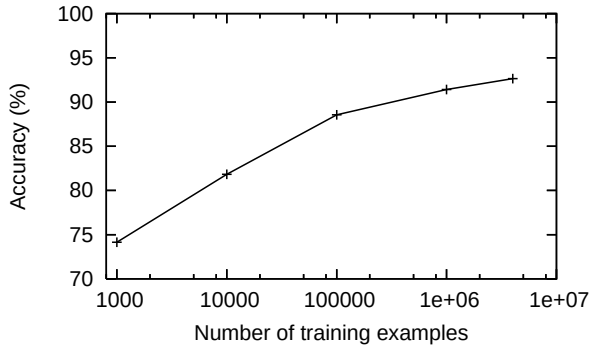

Figure 3: Noun gender classification learning curve for FULL-AUTO

does not appear destined to reach the performance obtained through other means described below.

We achieve even higher accuracy when the output of the -AUTO systems are combined with the original gender counts (the semi-supervised extension). The relative value of the context and type-based features is now reversed: using only context-based features (CONTEXT-SEMI) achieves 92.4%, while using only type-based features (TYPE-SEMI) achieves 91.3%. This is because much of the type information is already implicit in the PATHGENDER counts. The TYPE-AUTO predictions contribute little information, only fragmenting the data and leading to over-training and lower accuracy. On the other hand, the CONTEXT-AUTO predictions improve accuracy, as these scores provide orthogonal and hence helpful information for the semi-supervised classifier.

Combining FULL-AUTO with our enhanced path gender counts, PATHGENDER+, results in the overall best performance, 95.5% for FULL-SEMI, significantly better than PATHGENDER+ alone.[7] This is a 50% error reduction over the PATHGENDER system, strongly confirming the benefit of our semi-supervised approach.

To illustrate the importance of the unlabeled data, we created a system that uses all features, including the PATHGENDER+ counts, and trained this system using only the gold standard training data. This system was unable to leverage the extra features to improve performance; its accuracy was 92.0%, roughly equal to PATHGENDER+ alone. While SVMs work well with high-dimensional data, they simply cannot exploit features that do not occur in the training set.

5.2 Further improvements

We can improve performance further by doing some simple coreference before assigning gender. Currently, we only group nouns with the same string, and then decide gender collectively for the group. There are a few cases, however, where an ambiguous surname, such as *Willey*, can only be classified correctly if we link the surname to an earlier instance of the full name, e.g. *Katherine Willey*. We thus added the following simple post-processing rule: If a noun is classified as $masculine$ or $feminine$ (like the ambiguous *Willey*), and it was observed earlier as the last part of a larger noun, then re-assign the gender to $masculine$ or $feminine$ if one of these is the most common path-gender count for the larger noun. We back off to counts for the first name (e.g. *Kathleen* .*) if the full name is unobserved.

This enhancement improved the PATHGENDER and PATHGENDER+ systems to 93.3% and 94.3%, respectively, while raising the accuracy of our FULL-SEMI system to 96.7%. This demonstrates that the surname-matching post-processor is a simple but worthwhile extension to a gender predictor.[8]

The remaining errors represent a number of challenging cases: *United States*, *group*, and *public* labeled as $plural$ but classified as $neutral$; *spectator* classified as $neutral$, etc. Some of these may yield to more sophisticated joint classification of coreference and gender, perhaps along the lines of work in named-entity classification (Bunescu and Mooney, 2004) or anaphoricity (Denis and Baldridge, 2007).

While gender has been shown to be the key feature for statistical pronoun resolution (Ge et al., 1998; Bergsma and Lin, 2006), it remains to be seen whether the exceptional accuracy obtained here will translate into improvements in resolution performance. However, given the clear utility of gender in coreference, substantial error reductions in gender

[7] We evaluate significance using McNemar's test, $p<0.01$. Since McNemar's test assumes independent classifications, we apply the test to the classification of noun *groups*, not instances.

[8] One might wonder, why not provide special features so that the system can *learn* how to handle ambiguous nouns that occurred as sub-phrases in earlier names? The nature of our training data precludes this approach. We only include *un*ambiguous examples as pseudo-seeds in the learning process. Without providing ambiguous (but labeled) surnames in some way, the learner will not take advantage of features to help classify them.

assignment will likely be a helpful contribution.

6 Related Work

Most coreference and pronoun resolution papers mention that they use gender information, but few explain how it is acquired. Kennedy and Boguraev (1996) use gender information produced by their enhanced part-of-speech tagger. Gender mistakes account for 35% of their system's errors. Gender is less crucial in some genres, like computer manuals; most nouns are either neutral or plural and gender can be determined accurately based solely on morphological information (Lappin and Leass, 1994).

A number of researchers (Evans and Orăsan, 2000; Soon et al., 2001; Harabagiu et al., 2001) use WordNet classes to infer gender knowledge. Unfortunately, manually-constructed databases like WordNet suffer from both low coverage and rare senses. Pantel and Ravichandran (2004) note that the nouns *computer* and *company* both have a WordNet sense that is a hyponym of *person*, falsely indicating these nouns would be compatible with pronouns like *he* or *she*. In addition to using WordNet classes, Soon et al. (2001) assign gender if the noun has a gendered designator (like *Mr.* or *Mrs.*) or if the first token is present on a list of common human first names. Note that we incorporate such contextual and categorical information (among many other information sources) automatically in our discriminative classifier, while they manually specify a few high-precision rules for particular gender cues.

Ge et al. (1998) pioneered the statistical approach to gender determination. Like others, they consider gender and number separately, only learning statistical gender for the masculine, feminine, and neutral classes. While gender and number can be handled together for pronoun resolution, it might be useful to learn them separately for other applications. Other statistical approaches to English noun gender are discussed in Section 2.

In languages with 'grammatical' gender and plentiful gold standard data, gender can be tagged along with other word properties using standard supervised tagging techniques (Hajič and Hladká, 1997). While our approach is the first to exploit a dual or orthogonal representation of English noun gender, a bootstrapping approach has been applied to determining grammatical gender in other languages by Cucerzan and Yarowsky (2003). In their work, the two orthogonal views are: 1) the context of the noun, and 2) the noun's morphological properties. Bootstrapping with these views is possible in other languages where context is highly predictive of gender class, since contextual words like adjectives and determiners inflect to agree with the grammatical noun gender. We initially attempted a similar system for English noun gender but found context alone to be insufficiently predictive.

Bootstrapping is also used in general information extraction. Brin (1998) shows how to alternate between extracting instances of a class and inducing new instance-extracting patterns. Collins and Singer (1999) and Cucerzan and Yarowsky (1999) apply bootstrapping to the related task of named-entity recognition. Our approach was directly influenced by the hypernym-extractor of Snow et al. (2005) and we provided an analogous summary in Section 1. While their approach uses WordNet to label hypernyms in raw text, our initial labels are generated automatically. Etzioni et al. (2005) also require no labeled data or hand-labeled seeds for their named-entity extractor, but by comparison their classifier only uses a very small number of both features and automatically-generated training examples.

7 Conclusion

We have shown how noun-pronoun co-occurrence counts can be used to automatically annotate the gender of millions of nouns in unlabeled text. Training from these examples produced a classifier that clearly exceeds the state-of-the-art in gender classification. We incorporated thousands of useful but previously unexplored indicators of noun gender as features in our classifier. By combining the predictions of this classifier with the original gender counts, we were able to produce a gender predictor that achieves 95.5% classification accuracy on 2596 test nouns, a 50% reduction in error over the current state-of-the-art. A further name-matching post-processor reduced error even further, resulting in 96.7% accuracy on the test data. Our final system is the broadest and most accurate gender model yet created, and should be of value to many pronoun and coreference resolution systems.

References

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *COLING-ACL*, pages 33–40.

Shane Bergsma. 2005. Automatic acquisition of gender information for anaphora resolution. In *Canadian Conference on Artificial Intelligence*, pages 342–353.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology*, pages 172–183.

Razvan Bunescu and Raymond J. Mooney. 2004. Collective information extraction with relational Markov networks. In *ACL*, pages 438–445.

Eugene Charniak and Micha Elsner. 2009. EM works for pronoun anaphora resolution. In *EACL*.

Colin Cherry and Shane Bergsma. 2005. An expectation maximization approach to pronoun resolution. In *CoNLL*, pages 88–95.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *EMNLP-VLC*, pages 100–110.

Koby Crammer and Yoram Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292.

Silviu Cucerzan and David Yarowsky. 1999. Language independent named entity recognition combining morphological and contextual evidence. In *EMNLP-VLC*, pages 90–99.

Silviu Cucerzan and David Yarowsky. 2003. Minimally supervised induction of grammatical gender. In *NAACL*, pages 40–47.

Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

Pascal Denis and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference using integer programming. In *NAACL-HLT*, pages 236–243.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134.

Richard Evans and Constantin Orăsan. 2000. Improving anaphora resolution by identifying animate entities in texts. In *DAARC*, pages 154–162.

Michael Fleischman, Eduard Hovy, and Abdessamad Echihabi. 2003. Offline strategies for online question answering: answering questions before they are asked. In *ACL*, pages 1–7.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–171.

Andrew R. Golding and Dan Roth. 1999. A Winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3):107–130.

Jan Hajič and Barbora Hladká. 1997. Probabilistic and rule-based tagger of an inflective language: a comparison. In *ANLP*, pages 111–118.

Sanda Harabagiu, Razvan Bunescu, and Steven Maiorano. 2001. Text and knowledge mining for coreference resolution. In *NAACL*, pages 55–62.

Nancy Ide and Keith Suderman. 2004. The American National Corpus first release. In *LREC*, pages 1681–84.

Christopher Kennedy and Branimir Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *COLING*, pages 113–118.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Dekang Lin. 1998. Dependency-based evaluation of MINIPAR. In *LREC Workshop on the Evaluation of Parsing Systems*.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In *HLT-NAACL*, pages 321–328.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*, pages 1297–1304.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *ICML*.

Ellen Vorhees. 2002. Overview of the TREC 2002 question answering track. In *Proceedings of the Eleventh Text REtrieval Conference (TREC)*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, pages 189–196.