Phrase Structures and Dependencies for End-to-End Coreference Resolution

Anders Björkelund *Jonas Kuhn*
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
{anders,jonas}@ims.uni-stuttgart.de

ABSTRACT
We present experiments in data-driven coreference resolution comparing the effect of different syntactic representations provided as features in the coreference classification step: no syntax, phrase structure representations, dependency representations, and combinations of the representation types. We compare the end-to-end performance of a parametrized state-of-the-art coreference resolution system on the English data from the CoNLL 2012 shared task. On their own, phrase structures are more useful than dependencies, but the combinations yield highest performance and a significant improvement on the resolution of pronouns.
Enriching phrase structure with dependency trees obtained from an independent parser is most helpful, but an extension of the predicted phrase structure using just pattern-based phrase-to-dependency conversion seems to provide signals for the machine learning that cannot be distilled from phrase structure alone (despite intense feature selection). This is an interesting result for a highly configurational language: It is easier to learn generalizations over grammatical constraints on coreference when grammatical relations are explicitly provided.

KEYWORDS: Coreference Resolution, Dependency Parsing vs. Phrase-structure Parsing.

Proceedings of COLING 2012: Posters, pages 145–154,
COLING 2012, Mumbai, December 2012.

1 Introduction

Data-driven coreference resolution has received a lot of recent attention, including the 2011 and 2012 CoNLL shared tasks (Pradhan et al., 2011, 2012). To a greater or lesser extent, most coreference systems make use of syntax. For the subtask of mention detection, i.e., identifying referential phrases (substrings) for which coreference relations are subsequently determined, a phrase structure representation is useful for obvious reasons – in particular for the standard coreference task focusing on noun phrase (NP) and pronoun resolution. But also for the subsequent subtask, coreference resolution, syntactic information has proven useful in data-driven approaches – as one might expect from the rich linguistic work on Binding Theory, which targets the grammatical constraints on possible interpretations of referential phrases. It is this second subtask that we will parametrize systematically in this paper.

Most coreference work has built on phrase structure syntax, although dependency syntax was, for instance, used in the SemEval 2010 Task 1 (Recasens et al., 2010). To our knowledge, effects of the two main alternatives have not been studied systematically. The choice typically seems to be driven by external factors (such as availability in shared task data). The fact that mention detection is so straightforward with phrase structure input also creates a practical bias affecting the full pipeline, but since both the phrase structure and the dependency parsing research paradigms are at mature stages, with parsers available for many languages, a more informed decision would be desirable.

We here intend to shed some initial light on how the two different syntactic representations fare comparatively in end-to-end coreference resolution: What is the best basis for machine learning to pick up the (sometimes subtle) grammatical constraints influencing coreference resolution? Starting from a state-of-the-art system, we compare a phrase-structure-based resolver with a dependency-based counterpart and combinations of the two syntactic information sources on the English data from the CoNLL 2012 Shared Task. In a nutshell, the main results are that as a single source of information, phrase structures are more useful than dependencies, but experiments indicate that the two might be complementary: combined feature information from both sources outperform the phrase-structure-based system, particularly with respect to pronouns.

2 Grammatical Factors in Coreference Relations

For decades, coreference data have been at the core of many considerations (and debates) in Generative Linguistics, because grammatical configurations influence the availability of certain readings and hence make coreference tests a useful (albeit mostly theory-dependent) diagnostic for many linguistic purposes. Typical examples of facts addressed by Binding Theory are the following:

(1) a. John_i thinks that Bill_j hurt $\text{himself}_{*i/j}$.
    b. John_i thinks that Bill_j hurt $\text{him}_{i/*j}$.
    c. He_i hurt $\text{John}_{*i/j}$.

Roughly speaking, (A) reflexives like *himself* have to be coreferent with an element inside of their local clause, whereas (B) non-reflexive pronouns like *him* must have an antecedent outside of their local clause. (C) Full NPs, such as proper names, must not be preceded by a coreferent NP in the same sentence. Chomsky (1981) describes the grammatical constraints over possible coreference interpretations by three Binding Principles (A, B, C), which have been discussed, extended and criticized in countless contributions in the linguistic literature.

Given that there are grammatical constraints of this kind, one may expect that hard-coding some of the Binding Principles should help in practical coreference systems. However, the treatment of more subtle cases is quite controversial in the literature and sometimes involves fairly involved assumptions about phrase structure; in addition, there are a number of contextually driven or construction-specific exceptions to the grammar-driven principles, such as so-called logophoric usages of reflexives (2), and plain pronouns in contexts where one would expect reflexives (3) (examples due to (König and Gast, 2002)).

(2) Ronni$_i$ suspected that was probably true [...] [S]omething else [...] had provoked her$_i$ own furious outburst [...] Some more personal resentment that had come from within herself$_i$. [BNC JXT 2086]

(3) John did not have any money on him (/*himself).

In this light, a somewhat less committed but practically effective way is to provide the relevant "building blocks" of the Binding Principles as features for machine learning of the coreference relation, so the general principles (and possibly even some of the systematic exceptions) can be picked up from the training data. One may assume that this is in effect what happens when the inclusion of syntactic features in coreference classification leads to an improvement in accuracy. (Additionally, a trained system will react more gracefully to parsing errors.)

But what are the relevant building blocks of the Binding Principles that should be provided as syntactic features in coreference classification? Chomsky's original formulation relies on phrase-structural configurations, making reference to the so-called *governing category* of an anaphoric element: reflexive pronouns must be bound[1] within their governing category, whereas non-reflexive pronouns must be free (not bound) within their governing category. The governing category of some element X is defined as the minimal domain that includes X, X's governor (typically the element that subcategorize for X) and an accessible SUBJECT.[2] Any details are beyond the scope of this paper, suffice it to note that all relevant notions are ultimately defined with respect to phrase structure (following the full-fledged representations of Government-and-Binding Theory, in this case). So, in theory, phrase structure features alone should be sufficient input to machine learning.

Yet it is probably clear even from the brief exposition that the conditions underlying the principles are highly complex, so it is quite possible that even in an expressive machine learning paradigm with powerful feature selection, the relevant notions may be hard to pick up. We note that certain relational notions like subject play a central role. So, could it be helpful to offer a simple labeling of the grammatical relations as additional building blocks for the machine learning – even though it is in principle possible to derive these notions from the syntax tree?

In constraint-based approaches to syntax, Chomsky's purely phrase-structure-based approach has been criticized, and (Pollard and Sag, 1992) and (Dalrymple, 1993), among others, argue for alternative statements of the Binding Principles, using relational notions and referring to various prominence hierarchies.[3] So, according to these approaches, phrase structural configurations

[1] Binding is also defined with respect to phrase structure configuration: X **binds** Y, if X and Y are co-indexed (i.e., interpreted as coreferent), and X c-commands Y. (X is again defined to **c-command** Y, if X and Y do not dominate each other in the tree, and the first branching node dominating X also dominates Y).

[2] The notion of "accessible SUBJECT", as opposed to the plain notion of subject, takes care of subtle distinctions between tensed and untensed clauses and the role that possessives play; however it is ultimately defined configurationally as well.

[3] In (Dalrymple, 1993), e.g., Binding Principles are stated as a combination of an abstraction over grammatical function paths (following Lexical-Functional Grammar) and conditions on the ranking of the antecedent and the anaphor within a hierarchy of thematic roles.

are not the (only) relevant building blocks one should consider – even from the theoretical perspective. The results from an end-to-end evaluation of real-life coreference systems using off-the-shelf phrase structure and dependency parsers will of course by no means allow us to differentiate between the theoretical paradigms; but we believe that a systematic comparison will help increase awareness of how different syntactic paradigms emphasize different syntactic properties in their core representations and how this may affect downstream processing tasks.

3 Coreference System

We use our in-house coreference resolver (Björkelund and Farkas, 2012), which obtained the second best result in the CoNLL 2012 shared task. At the core, the system is similar to the pair-wise model proposed by Soon et al. (2001), which has become a *de facto* standard in coreference research during the last decade. However, the system features some extensions, including the use of multiple decoders that are combined through stacking. It also uses a rich feature set that includes both lexical information and syntax paths. The system is parametrized to allow for flexible experimentation with different feature sets. Since the system relies on a linear classifier, the parametrization also supports conjunctions between basic features.

The system works in thee stages: First, mentions are extracted by a set of rules that work on a phrase structure tree and extract all pronouns and noun phrases. Additionally, a statistical classifier is applied to filter out non-referential instances of certain pronouns (such as expletive *it*). The second stage is a cluster-based coreference algorithm that relies on a pairwise classifier. This resolver gives relatively small, but consistent clusters. The third stage is a standard best-first resolver (Ng and Cardie, 2002) that, in addition to the features used by the previous resolver, also encodes the *output* of the previous resolver into its feature space. For a more detailed description we refer to (Björkelund and Farkas, 2012).

The system relies on a phrase structure tree for two purposes: 1) For mention extraction; 2) As features for the pair-wise classifier. Since our systematic comparison focuses on the latter, we keep a phrase-structure-based mention extraction module fixed throughout the experiments.

Syntax-based features. To provide the "building blocks" for picking up machine-learned variants of the Binding Principles, we provide two types of feature templates building on the output of the parser: the first represents the *syntax path* in the phrase structure tree between two mentions. For example, consider the mentions "Kofi Annan" and "himself" in Figure 1. Here the path would be represented as `PRP↑NP↑VP↑VP↑S↓NP` from the anaphor to the antecedent.

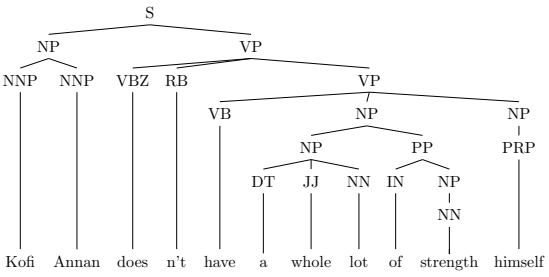

Figure 1: An example phrase structure tree.

Note that the path may provide some relevant characteristics of the structural "domain" that includes the reflexive and its (candidate) antecedent to mimic the Binding Principles: a reflexive needs to be bound within its *governing category*, and indeed the given path includes no major clause boundaries (no S) – but is there an accessible SUBJECT? The sub-path ↑S↓NP does reflect the subject configuration in English, but note that it will also occur for additional NPs like temporal ones as in *Last year, he left* or for topicalized NPs. Moreover, the tree paths aid the resolution algorithm in two ways: On the one hand, it may convince the pairwise classifier that two mentions in the same sentence are coreferent. On the other hand, it may also disallow coreference and prohibit false positive links when the antecedent is in a preceding sentence.

Now consider the dependency representation of the same sentence in Figure 2. With the dependency tree a corresponding path from the head of the anaphor to the head of the antecedent can be computed, i.e., ↑ADV↑VC↓SBJ. In this case, the grammatical function of the antecedent is explicitly captured in the syntax path. (Yet, from the dependency label path alone it may be hard to reliably identify the categorical characteristics of binding domains.)

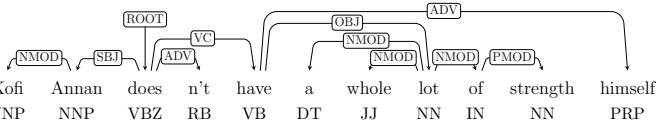

Figure 2: Dependency representation of the example from Figure 1.

Besides the path features, we also have feature templates that capture the local syntactic context of the mentions under consideration and of their immediate ancestors in the phrase structure tree. This can mimic a certain amount of subcategorization information or may indicate certain subclasses of mentions. For example, the local tree context of the antecedent NP can be described as `NP → NNP NNP`, and its ancestor tree context as `S → NP VP`. So for this example, configurationality of English actually indicates (implicitly) that the antecedent is, in fact, a subject. The local tree expansion of the NP mention alone is also helpful, for instance to detect bare plurals.

Similarly to how the idea of syntax paths can be transferred to the dependency representation, we also transfer the local tree context features. For instance, the dependency-based local tree context of the antecedent in Figure 2 can be described as `SBJ → NMOD`. And the local dependency tree context of the ancestor of the antecedent can be derived from the head of the head noun, i.e., `ROOT → SBJ ADV VC`.

Feature selection. Given the set of newly generated dependency-based feature templates, we perform an automatic feature selection procedure that evaluates new feature templates and conjunctions thereof. Specifically, we start from a seed set of templates and a pool of candidate templates (including conjunctions). We then run a greedy forward selection, where we evaluate the combination of the seed set with each of the templates from the candidate pool. In every iteration the template that contributes the most (according to some metric) is removed from the pool an inserted in the seed set. This process is repeated until the contribution of adding new feature templates is below a certain threshold. For the feature selection we optimized towards the CoNLL average (cf. Section 5 for details on evaluation metrics).

4 Data sets and Dependency conversion

In the experiments we use the English data from this year's CoNLL Shared Task (Pradhan et al., 2012). The data set comes from the OntoNotes project (Hovy et al., 2006) and features a multi-layer annotation that includes, among other things, syntax, named entities, and coreference. In the shared task, these additional annotation layers were available during training and testing as well. In the testing case, *only predicted* versions of the additional layers are provided, based on off-the-shelf tools that were trained on the training portion.

Since the official test set has not yet been released, we use the development set as test set. In order to do feature engineering, we partitioned the documents in the training set into two sets – 75% used for training and 25% used for evaluation of new features.

To study the role of dependency information vs. phrase structure information in coreference classification, we added two variants of dependency annotations to the training and development sets. In the first variant, we use the dependency parser by Bohnet (2010), trained on the OntoNotes parse trees run through the phrase-to-dependency conversion of Choi and Palmer (2010). This conversion (henceforth Choi) takes advantage of the function labels in the phrase structure annotation and produces a rich label set. For instance, subjects and objects are distinguished by distinct dependency relations. In the same manner that the shared task data was prepared, we created predicted dependency trees for both the training (using 10-fold cross-validation) and the development sets using the Bohnet dependency parser (Bohnet, 2010).[4]

For the second variant, we created dependency trees automatically by converting the *predicted* phrase structure trees that are provided in the CoNLL data set using the Stanford conversion (de Marneffe et al., 2006), which uses rules for identifying phrase structure patterns for particular grammatical relations, taking advantage of the configurationality of English. Since these trees are converted from the predicted phrase structure trees, they are more likely to be synchronized with the NPs that are used as mentions, i.e., NPs are more likely to form proper subtrees in the dependency tree.

In conclusion, we experiment with three different syntactic annotations that are all predicted on the test set: 1) Predicted phrase structure trees from the CoNLL 2012 Shared Task; 2) Dependency trees obtained via the Stanford conversion when applied to the parse trees from 1); 3) Dependency trees obtained from the Bohnet parser that was trained on the Choi conversion of the OntoNotes parse trees.

5 Experimental Setup and Results

For the experiments we built 5 different systems that differ only in their feature representation:

1. Baseline (BL) – Our system (Björkelund and Farkas, 2012) stripped of all syntax-based features;
2. Reference (BL+PS) – Same as above, but including the syntax-based features, i.e., the same system as in (Björkelund and Farkas, 2012);
3. Choi dependencies (BL+DT_{Choi}) – The Baseline feature set, extended with dependency features from a dependency parser (Choi-style);
4. Choi dependencies and phrase structures (BL+PS+DT_{Choi}) – The Reference feature set, extended with Choi-style dependency features;

[4] Downloaded from `http://code.google.com/p/mate-tools/`

5. Stanford dependencies and phrase structures (BL+PS+DT$_{Stanf}$) – The Reference feature set, extended with dependency features from the rule-based Stanford conversion.

For systems 3, 4, and 5, the extended feature sets were computed by the automatic feature selection procedure describe above. The baseline provides a lower bound on how well coreference resolution can be accomplished without syntax-based features. Besides the baseline, system 3 is the only one that does not make use of phrase-structure-based features. Hence, this system will reveal the importance of phrase-structure-based features. Systems 4 and 5 allow us to measure if the combination of features from both syntactic paradigms improves the performance of the system. Finally, system 2 is a purely phrase-structure-based system with an already optimized feature set. This is the *reference system*, and it provides an upper bound for using the standard CoNLL annotation layers alone (i.e., not using any dependency-based features).[5]

Results. To evaluate the systems we use the official CoNLL scorer,[6] which computes several metrics including MUC (Vilain et al., 1995), BCUB (Bagga and Baldwin, 1998), and CEAF (Luo, 2005). For completeness we also present end-to-end mention detection (MD) F-measure and the CoNLL average, i.e., the unweighted arithmetic mean of MUC, BCUB and the entity-based CEAF (CEAFE). To avoid clutter, and since precision and recall do not provide additional insights for the discussion at hand, we only present the F-measures of the corresponding metrics. The results of all systems on the CoNLL development set are presented in Table 1.

Sys.	Feature set	MD	MUC	BCUB	CEAFE	CoNLL
1	BL	73.64	65.64	70.45	45.43	60.51
2	BL+PS	74.96	67.12	71.18	46.84	61.71
3	BL+DT$_{Choi}$	74.54	66.74	70.98	46.50	61.42
4	BL+PS+DT$_{Choi}$	75.23	67.69	71.48	47.02	62.07
5	BL+PS+DT$_{Stanf}$	75.23	67.46	71.22	47.18	61.96

Table 1: Results on coreference task when varying the feature set.

The results indicate that syntax-based features play an important role when it comes to resolving coreference. The baseline system, which does not use syntax in its feature set at all, is outperformed by the all other systems by more than a point in almost all metrics. The difference for all metrics is significant ($p < 0.005$).[7] Systems 2, 4, and 5 are all also significantly better than system 3 ($p < 0.05$). The systems that use a combination of both phrase-structure-based and dependency-based features obtain the highest scores, however compared to system 2, only the improvement in MUC for system 4 is significant ($p < 0.05$).

Error analysis. General quantitative error analysis for end-to-end coreference resolution is difficult, owing to the fact that the problem is ultimately a matter of evaluating partitionings over sets that do not necessarily contain the same elements. However, manual inspection of the alternative system outputs indicated that the systems using the combined feature set appeared to be better at finding the correct antecedent for pronouns. A crude quantitative analysis is to look at the links between a pronoun mention and its closest antecedent in the system output vs. the gold standard. While link-based metrics for coreference resolution have been criticized (see e.g. Luo (2005)), we believe that for pronouns they can still be an analytical device, since their antecedents tend to be close.

[5]The system and feature templates are available at http://www.ims.uni-stuttgart.de/~anders

[6]Downloaded from http://conll.cemantix.org/2012/

[7]Using a paired t-test over the documents

Specifically, for every pronoun in the gold standard, we regard the system output to be correct if (i) the nearest predicted antecedent to the left belongs to the same cluster as the mention in the gold standard; or (ii) if the mention is not part of a cluster in both the gold standard and the system output.[8] Otherwise the system prediction is regarded as incorrect. Based on these definitions, we computed the pronoun accuracy and broke down the results by by pronoun type, as shown in Table 2. The bottom-most row shows the total number of occurrences of each type.

System	Feature set	Standard	Possessive	Reflexive	All
1	BL	68.47	68.65	69.07	68.51
2	BL+PS	69.35	71.00	68.04	69.64
3	BL+DT$_{Choi}$	68.95	69.86	65.98	69.09
4	BL+PS+DT$_{Choi}$	70.00	71.63	74.23	70.35
5	BL+PS+DT$_{Stanf}$	69.51	71.69	69.07	69.91
Total		7,497	1,745	97	9,339

Table 2: Accuracy on pronouns.

The trends are similar to the improvement in the general coreference metrics. The difference between the non-syntax-based baseline system (1) and the reference system (2) is for all pronouns about 1% absolute. Note however that the improvement from system 2 to system 4 is not far behind with 0.7% absolute. This improvement is statistically significant ($p < 0.005$), as well as the improvement of system 5 over system 2 ($p < 0.05$). Our interpretation is that the small improvement in the coreference metrics (cf. Table 1) stems mostly from improved handling of pronouns.

6 Discussion and Conclusion

Starting out from a state-of-the-art coreference system for English, we experimented with phrase structure vs. dependency features for coreference resolution, studying effects on end-to-end performance (as shown in Table 1). On their own, dependencies (as in system 3) are a significantly weaker source of information than phrase structure (as in system 2) for coreference resolution in English. This is not too surprising since certain characteristics of grammatical binding domains are not captured in the latter system's dependency path information.

It also seems like like information from phrase structure and dependencies is orthogonal: although not significant overall, a combination yields better results (as in systems 4 and 5) than using phrase structures alone (system 2). System 4, with its independently obtained phrase structure and dependency structure, has the best performance overall according to most end-to-end metrics, and significantly so for the accuracies on pronoun links (compare Table 2).

It is worth noting that system 5, which uses "just" configurational patterns to identify and label grammatical relations in the predicted phrase structures already present in system 2, outperforms the latter according to all metrics. This means that the phrase-to-dependency conversion seems to add signals to the data that the system's machine learning cannot distill from phrase structure alone – despite intense feature selection. This is an interesting result for English as a highly configurational language: It is easier to learn generalizations over grammatical constraints on coreference when grammatical relations are explicitly provided. It can be expected that for other, less configurational languages, an even more pronounced difference can be observed. We plan to study this in future work.

[8] We ignore cataphoric pronouns since they do not have any antecedents to the left and it is not obvious how to include these in the evaluation. These cases are, however, rare and account for only about 3% of the pronouns in the test set.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) via the SFB 732 "Incremental Specification in Context", project D8.

References

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

Björkelund, A. and Farkas, R. (2012). Data-driven multilingual coreference resolution using resolver stacking. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 49–55, Jeju Island, Korea. Association for Computational Linguistics.

Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China. Coling 2010 Organizing Committee.

Choi, J. D. and Palmer, M. (2010). Robust Constituent-to-Dependency Conversion for English. In *Proceedings of 9th Treebanks and Linguistic Theories Workshop (TLT)*, pages 55–66.

Chomsky, N. (1981). *Lectures on Government and Binding*. Foris Publications, Dordrecht.

Dalrymple, M. (1993). *The Syntax of Anaphoric Binding*. CSLI Publications, Stanford, CA.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-2006*, pages 449–454, Genoa, Italy.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

König, E. and Gast, V. (2002). Reflexive pronouns and other uses of self-forms in English. *Zeitschrift für Anglistik und Amerikanistik*, 50(3):1–14.

Luo, X. (2005). On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Ng, V. and Cardie, C. (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pollard, C. and Sag, I. A. (1992). Anaphors in English and the scope of binding theory. *Linguistic Inquiry*, 23(2):261–303.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., and Xue, N. (2011). Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA. Association for Computational Linguistics.

Recasens, M., Màrquez, L., Sapena, E., Martí, M. A., Taulé, M., Hoste, V., Poesio, M., and Versley, Y. (2010). Semeval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 1–8, Uppsala, Sweden. Association for Computational Linguistics.

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

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A model theoretic coreference scoring scheme. In *Proceedings of the Sixth Conference on Message Understanding (MUC-6)*, Columbia, Maryland.