Visualization, Search, and Error Analysis for Coreference Annotations

Markus Gärtner   Anders Björkelund   Gregor Thiele   Wolfgang Seeker   Jonas Kuhn
Institute for Natural Language Processing
University of Stuttgart
`{thielegr,seeker,gaertnms,anders,kuhn}@ims.uni-stuttgart.de`

Abstract

We present the ICARUS Coreference Explorer, an interactive tool to browse and search coreference-annotated data. It can display coreference annotations as a tree, as an entity grid, or in a standard text-based display mode, and lets the user switch freely between the different modes. The tool can compare two different annotations on the same document, allowing system developers to evaluate errors in automatic system predictions. It features a flexible search engine, which enables the user to graphically construct search queries over sets of documents annotated with coreference.

1 Introduction

Coreference resolution is the task of automatically grouping references to the same real-world entity in a document into a set. It is an active topic in current NLP research and has received considerable attention in recent years, including the 2011 and 2012 CoNLL shared tasks (Pradhan et al., 2011; Pradhan et al., 2012).

Coreference relations are commonly represented by sets of mentions, where all mentions in one set (or coreference **cluster**) are considered coreferent. This type of representation does not support any internal structure within the clusters. However, many automatic coreference resolvers establish links between pairs of mentions which are subsequently transformed to a cluster by taking the transitive closure over all links, i.e., placing all mentions that are directly or transitively classified as coreferent in one cluster. This is particularly the case for several state-of-the-art resolvers (Fernandes et al., 2012; Durrett and Klein, 2013; Björkelund and Kuhn, 2014). These pairwise decisions, which give rise to a clustering, can be exploited for detailed error analysis and more fine-grained search queries on data automatically annotated for coreference.

We present the ICARUS Coreference Explorer (ICE), an interactive tool to browse and search coreference-annotated data. In addition to standard text-based display modes, ICE features two other display modes: an **entity-grid** (Barzilay and Lapata, 2008) and a **tree** view, which makes use of the internal pairwise links within the clusters. ICE builds on ICARUS (Gärtner et al., 2013), a platform for search and exploration of dependency treebanks.[1]

ICE is geared towards two (typically) distinct users: The *NLP developer* who designs coreference resolution systems can inspect the predictions of his system using the three different display modes. Moreover, ICE can compare the predictions of a system to a gold standard annotation, enabling the developer to inspect system errors interactively. The second potential user is the *corpus linguist*, who might be interested in browsing or searching a document, or a (large) set of documents for certain coreference relations. The built-in search engine of ICARUS now also allows search queries over sets of documents in order to meet the needs of this type of user.

2 Data Representation

ICE reads the formats used in the 2011 and 2012 CoNLL shared tasks as well as the SemEval 2010 format (Recasens et al., 2010).[2] Since these formats cannot accommodate pairwise links, an auxiliary file with standoff annotation can be provided, which we call *allocation*. An allocation is a list of pairwise links between mentions. Multiple

[1]ICE is written in Java and is therefore platform independent. It is open source (under GNU GPL) and we provide both sources and binaries for download on `http://www.ims.uni-stuttgart.de/data/icarus.html`

[2]These two formats are very similar tabular formats, but differ slightly in the column representations.

allocations can be associated with a single document and the user can select one of these for display or search queries. An allocation can also include *properties* on mentions and links. The set of possible properties is not constrained, and the user can freely specify properties as a list of key-value pairs. Properties on mentions may include, e.g., grammatical gender or number, or information status labels. Additionally, a special property that indicates the head word of a mention can be provided in an allocation. The head property enables the user to access head words of mentions for display or search queries.

The motivation for keeping the allocation file separate from the CoNLL or SemEval files is twofold: First, it allows ICE to work without having to provide an allocation file, thereby making it easy to use with the established formats for coreference. The user is still able to introduce additional structure by the use of the allocation file. Second, multiple allocation files allow the user to switch between different allocations while exploring a set of documents. Moreover, as we will see in Section 3.3, ICE can also compare two different allocations in order to highlight the differences.

In addition to user-specified allocations, ICE will always by default provide an internal structure for the clusters, in which the correct antecedent of every mention is the closest coreferent mention with respect to the linear order of the document (this is equivalent to the training instance creation heuristic proposed by Soon et al. (2001)). Therefore, the user is not required to define an allocation on their own.

3 Display Modes

In this section we describe the entity grid and tree display modes by means of screenshots. ICE additionally includes a standard text-based view, similar to other coreference visualization tools. The example document is taken from the CoNLL 2012 development set (Pradhan et al., 2012) and we use two allocations: (1) the predictions output by Björkelund and Kuhn (2014) system (*predicted*) and (2) a gold allocation that was obtained by running the same system in a restricted setting, where only links between coreferent mentions are allowed (*gold*). The complete document can be seen in the lower half of Figure 1.

3.1 Entity grid

Barzilay and Lapata (2008) introduce the *entity grid*, a tabular view of entities in a document. Specifically, rows of the grid correspond to sentences, and columns to entities. The cells of the table are used to indicate that an entity is mentioned in the corresponding sentence. Entity grids provide a compact view on the distribution of mentions in a document and allow the user to see how the description of an entity changes from mention to mention.

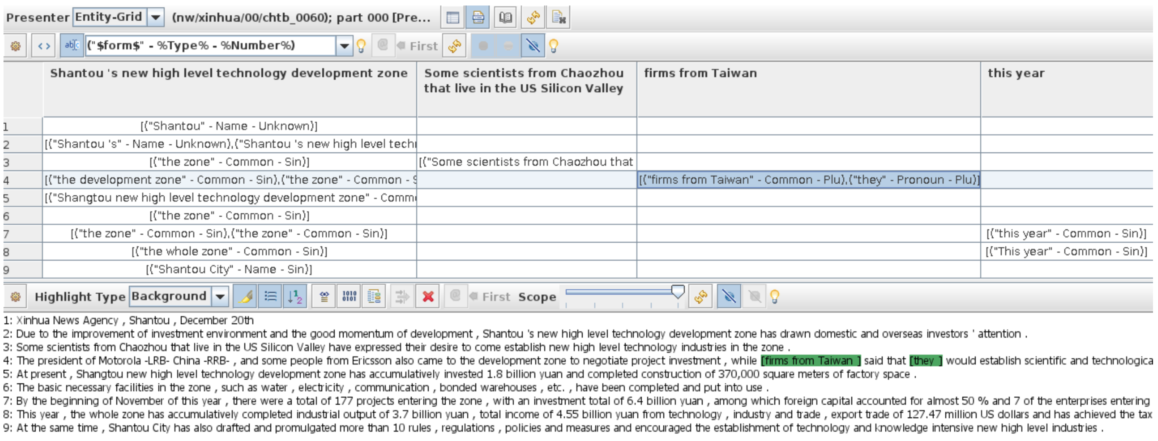
Figure 1 shows ICE's entity-grid view for the example document using the predicted allocation. When clicking on a cell in the entity grid the immediate textual context of the cell is shown in the lower pane. In Figure 1, the cell with the blue background has been clicked, which corresponds to the two mentions *firms from Taiwan* and *they*. These mentions are thus highlighted in the lower pane. The user can also right-click on a cell and jump straight to the tree view, centered around the same mentions.

3.2 Label Patterns

The information that is displayed in the cells of the entity grid (and also on the nodes in the tree view, see Section 3.3) can be fully customized by the user. The customization is achieved by defining *label patterns*. A label pattern is a string that specifies the format according to which a mention will be displayed. The pattern can extract information on a mention according to three axes: (1) at the token- level for the full mention, extracting, e.g., the sequence of surface forms or the part-of-speech tags of a mention; (2) at the mention- level, extracting an arbitrary property of a mention as defined in an allocation; (3) token-level information from the head word of a mention.

Label patterns can be defined interactively while displaying a document and the three axes are referenced by dedicated operators. For instance, the label pattern `$form$` extracts the full surface form of a mention, whereas `#form#` only extracts the surface form of the head word of a mention. All properties defined by the user in the allocation (see Section 2) are accessible via label patterns.

For example, the allocations we use for Figure 1 include a number of properties on the mentions, most of which are internally computed by the coreference system: The TYPE of a mention, which can take any of the values



Presenter Entity-Grid (nw/xinhua/00/chtb_0060); part 000 [Pre...

("$form$" - %Type% - %Number%)

	Shantou 's new high level technology development zone	Some scientists from Chaozhou that live in the US Silicon Valley	firms from Taiwan	this year
1	[{"Shantou" - Name - Unknown}]			
2	[{"Shantou 's" - Name - Unknown},{"Shantou 's new high level techn			
3	[{"the zone" - Common - Sin}]	[{"Some scientists from Chaozhou that		
4	[{"the development zone" - Common - Sin},{"the zone" - Common - S		[{"firms from Taiwan" - Common - Plu},{"they" - Pronoun - Plu}]	
5	[{"Shangtou new high level technology development zone" - Comm			
6	[{"the zone" - Common - Sin}]			
7	[{"the zone" - Common - Sin},{"the zone" - Common - Sin}]			[{"this year" - Common - Sin}]
8	[{"the whole zone" - Common - Sin}]			[{"This year" - Common - Sin}]
9	[{"Shantou City" - Name - Sin}]			

Highlight Type Background

```
1: Xinhua News Agency , Shantou , December 20th
2: Due to the improvement of investment environment and the good momentum of development , Shantou 's new high level technology development zone has drawn domestic and overseas investors ' attention .
3: Some scientists from Chaozhou that live in the US Silicon Valley have expressed their desire to come establish new high level technology industries in the zone .
4: The president of Motorola -LRB- China -RRB- , and some people from Ericsson also came to the development zone to negotiate project investment , while [firms from Taiwan ] said that [they ] would establish scientific and technologica
5: At present , Shangtou new high level technology development zone has accumulatively invested 1.8 billion yuan and completed construction of 370,000 square meters of factory space .
6: The basic necessary facilities in the zone , such as water , electricity , communication , bonded warehouses , etc. , have been completed and put into use .
7: By the beginning of November of this year , there were a total of 177 projects entering the zone , with an investment total of 6.4 billion yuan , among which foreign capital accounted for almost 50 % and 7 of the enterprises entering
8: This year , the whole zone has accumulatively completed industrial output of 3.7 billion yuan , total income of 4.55 billion yuan from technology , industry and trade , export trade of 127.47 million US dollars and has achieved the tax
9: At the same time , Shantou City has also drafted and promulgated more than 10 rules , regulations , policies and measures and encouraged the establishment of technology and knowledge intensive new high level industries .
```

Figure 1: Entity grid over the predicted clustering in the example document.

{Name, Common, Pronoun} and is inferred from the part- of-speech tags in the CoNLL file; The grammatical NUMBER of a mention, which is assigned based on the number and gender data compiled by Bergsma and Lin (2006) and can take the values {Sin, Plu, Unknown}. The label pattern for displaying the number property associated with a mention would be %Number%.

The label pattern used in Figure 1 is defined as ("$form$" - %Type% - %Number%). This pattern accesses the full surface form of the mentions ($form$), as well as the TYPE (%Type%) and grammatical NUMBER (%Number%) properties defined in the allocation file.

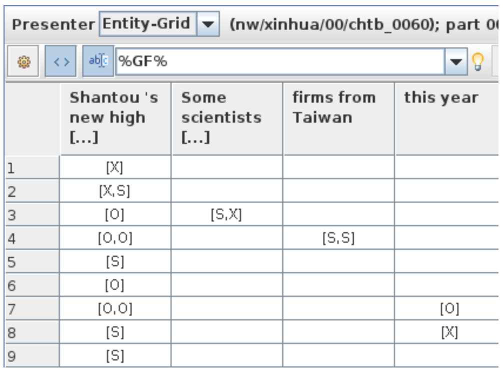
Custom properties and label patterns can be used for example to display the entity grid in the form proposed by Barzilay and Lapata (2008): In the allocation, we assign a coarse-grained grammatical function property (denoted GF) to every mention, where each mention is tagged as either *subject*, *object*, or *other* (denoted S, O, X, respectively).[3] The label pattern %GF% then displays the grammatical function of each mention in the entity grid, as shown in Figure 2.

3.3 Tree view

Pairwise links output by an automatic coreference system can be treated as arcs in a directed graph. Linking the first mention of each cluster to an artificial root node creates a tree structure that encodes the entire clustering in a document. This representation has been used in coreference resolvers (Fernandes et al., 2012; Björkelund and Kuhn, 2014), but ICE uses it to display links between mentions introduced by an automatic (pairwise) resolver.

Figure 3 shows three examples of the tree view of the same document as before: The gold allocation (3a), the predicted allocation (3b), as well as the differential view, where the two allocations are compared (3c). Each mention corresponds to a node in the trees and all mentions are directly or transitively dominated by the artificial root node. Every subtree under the root constitutes its own cluster and a *solid* arc between two mentions denotes that the two mentions are coreferent according to a coreference allocation. The information displayed in the nodes of the tree can be customized using label patterns.

In the differential view (Figure 3c), solid arcs correspond to the predicted allocation. Dashed nodes and arcs are present in the gold allocation, but not in the prediction. Discrepancies between the predicted and the gold allocations are marked



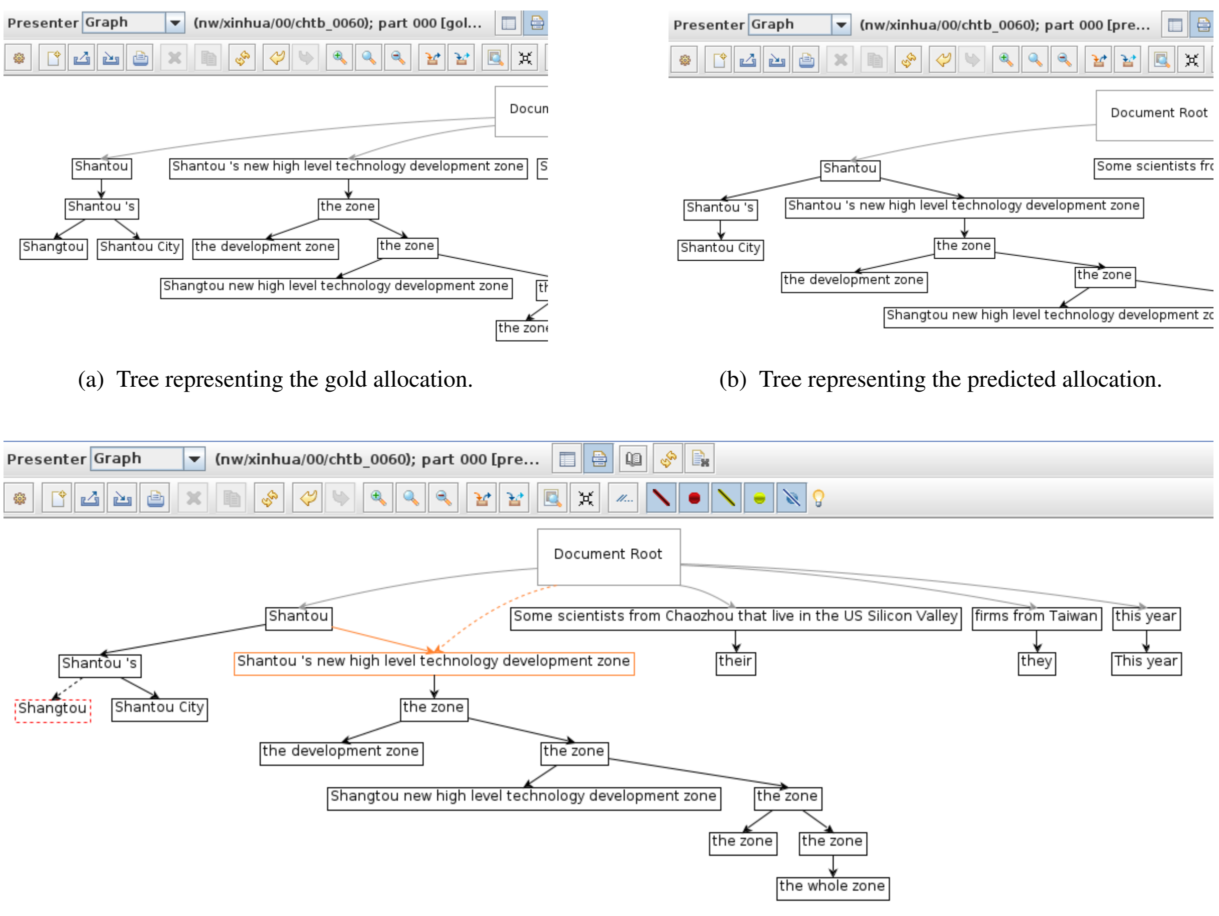
Presenter Entity-Grid (nw/xinhua/00/chtb_0060); part 0

%GF%

	Shantou 's new high [...]	Some scientists [...]	firms from Taiwan	this year
1	[X]			
2	[X,S]			
3	[O]	[S,X]		
4	[O,O]		[S,S]	
5	[S]			
6	[O]			
7	[O,O]			[O]
8	[S]			[X]
9	[S]			

Figure 2: Example entity grid, using the labels by Barzilay and Lapata (2008).

[3]The grammatical function was assigned by converting the phrase-structure trees in the CoNLL file (which lack grammatical function information) to Stanford dependencies (de Marneffe and Manning, 2008), and then extracting the grammatical function from the head word in each mention.


Presenter Graph
(nw/xinhua/00/chtb_0060); part 000 [gol...
(nw/xinhua/00/chtb_0060); part 000 [pre...
Document Root
Shantou
Shantou 's
Shangtou
Shantou City
Shantou 's new high level technology development zone
Some scientists from Chaozhou that live in the US Silicon Valley
firms from Taiwan
this year
their
they
This year
the zone
the development zone
Shangtou new high level technology development zone
the whole zone

(a) Tree representing the gold allocation.

(b) Tree representing the predicted allocation.

(c) Differential view displaying the difference between the gold and predicted allocations.

Figure 3: Tree view over the example document (gold, predicted, differential).

with different colors denoting different types of errors. The example in Figure 3c contains two errors made by the system:

1. A **false negative mention**, denoted by the dashed red node *Shangtou*. In the gold standard (Figure 3a) this mention is clustered with other mentions such as *Shantou 's*, *Shantou City*, etc. The dashed arc between *Shantou 's* and *Shangtou* is taken from the gold allocation, and indicates what the system prediction should have been like.[4]

2. A **foreign antecedent**, denoted by the solid orange arc between *Shantou 's new high level technology development zone* and *Shantou*. In this case, the coreference system erroneously clustered these two mentions. The correct antecedent is indicated by the dashed arc that originates from the document root.

[4]This error likely stems from the fact that *Shantou* is spelled two different ways within the same document which causes the resolver's string-matching feature to fail.

This error is particularly interesting since the system effectively merges the two clusters corresponding to *Shantou* and *Shantou' s new high level technology development zone*. The tree view, however, shows that the error stems from a single link between these two mentions, and that the developer needs to address this.

Since the tree-based view makes pairwise decisions explicit, the differential view shown in Figure 3c is more informative to NLP developers when inspecting errors by automatic system than comparing a gold standard clustering to a predicted one. The problem with analyzing the error on clusterings instead of trees is that the clusters would be merged, i.e., it is not clear where the actual mistake was made.

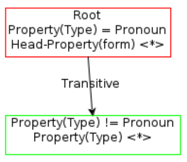
Additional error types not illustrated by Figure 3c include **false positive** mentions, where the system invents a mention that is not part of the gold allocation. When a false positive mention is assigned as an antecedent of another mention, the corresponding link is marked as an **invented antecedent**. Links that erroneously start a new cluster when it is coreferent with other mentions to the left is marked as **false new**.

4 Searching

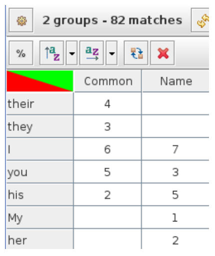
The search engine in ICE makes the annotations in the documents searchable for, e. g., a corpus linguist who is interested in specific coreference phenomena. It allows the user to express queries over mentions related through the tree. Queries can access the different layers of annotation, both from the allocation file and the underlying document, using various constructs such as, e.g., transitivity, regular expressions, and/or disjunctions. The user can construct queries either textually (through a query language) or graphically (by creating nodes and configuring constraints in dialogues). For a further discussion of the search engine we refer to the original ICARUS paper (Gärtner et al., 2013).

Figure 4 shows a query that matches cataphoric pronouns, i.e., pronouns that precede their antecedents. The figure shows the query expressed as a subgraph (on the left) and the corresponding results (right) obtained on the development set of the English CoNLL 2012 data using the manual annotation represented in the gold allocation.

The query matches two mentions that are directly or transitively connected through the graph. The first mention (red node) matches mentions of the type `Pronoun` that have to be attached to the document root node. In the tree formalism we adopt, this implies that it must be the first mention of its cluster. The second mention (green node) matches any mention that is not of the type `Pronoun`.



(a)



🟥🟩	Common	Name
their	4	
they	3	
I	6	7
you	5	3
his	2	5
My		1
her		2

(b)

Figure 4: Example search query and corresponding results.

The search results are grouped along two axes: the surface form of the head word of the first (red) node, and the type property of the second mention (green node), indicated by the special *grouping operator* `<*>` inside the boxes. The corresponding results are shown in the right half of Figure 4, where the first group (surface form) runs vertically, and the second group (mention type) runs horizontally. The number of hits for each configuration is shown in the corresponding cell. For example, the case that the first mention of a chain is the pronoun *I* and the closest following coreferent mention that is not a pronoun is of type `Common`, occurs 6 times. By clicking on a cell, the user can jump straight to a list of the matches, and browse them using any of the three display modes.

5 Related Work

Two popular annotation and visualization tools for coreference are PAlinkA (Orăsan, 2003) and MMAX2 (Müller and Strube, 2006), which focus on a (customizable) textual visualization with highlighting of clusters. The TrED (Pajas and Štěpánek, 2009) project is a very flexible multi-level annotation tool centered around tree-based annotations that can be used to annotate and visualize coreference. It also features a powerful search engine. Recent annotation tools include the web-based BRAT (Stenetorp et al., 2012) and its extension WebAnno (Yimam et al., 2013). A dedicated query and exploration tool for multi-level annotations is ANNIS (Zeldes et al., 2009).

The aforementioned tools are primarily meant as annotation tools. They have a tendency of locking the user into one type of visualization (tree- or text-based), while often lacking advanced search functionality. In contrast to them, ICE is not meant to be yet another annotation tool, but was designed as a dedicated coreference exploration tool, which enables the user to swiftly switch between different views. Moreover, none of the existing tools provide an entity-grid view.

ICE is also the only tool that can graphically compare predictions of a system to a gold standard with a fine-grained distinction on the types of differences. Kummerfeld and Klein (2013) present an algorithm that transforms a predicted coreference clustering into a gold clustering and records the necessary transformations, thereby quantifying different types of errors. However, their algorithm only works on clusterings (sets of mentions), not pairwise links, and is therefore not able to pinpoint some of the mistakes that ICE can (such as the foreign antecedent described in Section 3).

6 Conclusion

We presented ICE, a flexible coreference visualization and search tool. The tool complements standard text-based display modes with entity-grid and tree visualizations. It is also able to display discrepancies between two different coreference annotations on the same document, allowing NLP developers to debug coreference systems in a graphical way. The built-in search engine allows corpus linguists to construct complex search queries and provide aggregate result views over large sets of documents. Being based on the ICARUS platform's plugin-engine, ICE is extensible and can easily be extended to cover additional data formats.

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research (BMBF) via CLARIN-D, No. 01UG1120F and the German Research Foundation (DFG) via the SFB 732, project D8.

References

Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34(1):1–34.

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

Anders Björkelund and Jonas Kuhn. 2014. Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features. In *ACL*, Baltimore, MD, USA, June.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The stanford typed dependencies representation. In *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*.

Greg Durrett and Dan Klein. 2013. Easy Victories and Uphill Battles in Coreference Resolution. In *EMNLP*, pages 1971–1982, Seattle, Washington, USA, October.

Eraldo Fernandes, Cícero dos Santos, and Ruy Milidiú. 2012. Latent Structure Perceptron with Feature Induction for Unrestricted Coreference Resolution. In *EMNLP-CoNLL: Shared Task*, pages 41–48, Jeju Island, Korea, July.

Markus Gärtner, Gregor Thiele, Wolfgang Seeker, Anders Björkelund, and Jonas Kuhn. 2013. ICARUS – An Extensible Graphical Search Tool for Dependency Treebanks. In *ACL: System Demonstrations*, pages 55–60, Sofia, Bulgaria, August.

Jonathan K. Kummerfeld and Dan Klein. 2013. Error-Driven Analysis of Challenges in Coreference Resolution. In *EMNLP*, pages 265–277, Seattle, Washington, USA, October.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang.

Constantin Orăsan. 2003. PALinkA: A highly customisable tool for discourse annotation. In Akira Kurematsu, Alexander Rudnicky, and Syun Tutiya, editors, *Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue*, pages 39–43.

Petr Pajas and Jan Štěpánek. 2009. System for Querying Syntactically Annotated Corpora. In *ACL-IJCNLP: Software Demonstrations*, pages 33–36, Suntec, Singapore.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. In *CoNLL: Shared Task*, pages 1–27, Portland, Oregon, USA, June.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. In *EMNLP-CoNLL: Shared Task*, pages 1–40, Jeju Island, Korea, July.

Marta Recasens, Lluís Màrquez, Emili Sapena, M. Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 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, July.

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.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *EACL: Demonstrations*, pages 102–107, April.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In *ACL: System Demonstrations*, pages 1–6, August.

Amir Zeldes, Julia Ritz, Anke Lüdeling, and Christian Chiarcos. 2009. ANNIS: a search tool for multi-layer annotated corpora. In *Proceedings of Corpus Linguistics*.