SinoCoreferencer: An End-to-End Chinese Event Coreference Resolver

Chen Chen and Vincent Ng

Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{yzcchen,vince}@hlt.utdallas.edu

Abstract

This paper describes the design, implementation, and evaluation of SinoCoreferencer, a publicly-available end-to-end ACE-style Chinese event coreference system that achieves state-of-the-art performance on the ACE 2005 corpus. SinoCoreferencer comprises eight information extraction system components, including those for entity extraction, entity coreference resolution, and event extraction. Its modular design makes it possible to run each component in a standalone manner, thus facilitating the development of high-level Chinese natural language applications that make use of any of these core information extraction components. To our knowledge, SinoCoreferencer is the first publicly-available Chinese event coreference resolution system.

Keywords: coreference, information extraction, Chinese

1. Introduction

Event coreference resolution is the task of determining which event mentions in a text refer to the same real-world event. Its well-known entity counterpart, entity coreference resolution, is the task of determining which entity mentions in a text refer to the same real-world entity. While entity coreference is considered one of the most difficult tasks in natural language processing (NLP) (Mitkov et al., 2001), event coreference is arguably even more challenging: an event coreference resolver is typically situated at the end of the information extraction pipeline, assuming as input the outputs produced by various text-processing components. Compared to entity coreference, there is relatively little work on event coreference. In fact, almost all recent work on event coreference has reported results for English (e.g., Humphreys et al. (1997), Chen et al. (2009), Bejan and Harabagiu (2010), Chen et al. (2010), Chen et al. (2011), Lee et al. (2012)).

Our goal in this paper is to present the design and implementation of SinoCoreferencer, a publicly-available end-to-end ACE-style Chinese event coreference system that achieves state-of-the-art performance on the ACE 2005 corpus. SinoCoreferencer comprises eight information extraction system components, including those for entity extraction, entity coreference resolution, and event extraction. Its modular design makes it possible to run each component in a standalone manner, thus facilitating the development of high-level Chinese natural language applications that make use of any of these core information extraction components. To our knowledge, SinoCoreferencer is the first publicly-available Chinese event coreference resolution system.[1]

2. Design and Implementation

This section describes the design and implementation of SinoCoreferencer. SinoCoreferencer takes raw text as input and first uses the Stanford CoreNLP tool[2] to perform various kinds of preprocessing, including sentence segmentation, word tokenization, part-of-speech tagging and syntactic parsing.

After preprocessing, the text then passes through the eight components of SinoCoreferencer, which are shown in Figure 1. As we can see, the eight components can be roughly divided into four subsystems, one for entity extraction, one for entity coreference, one for event extraction, and one for event coreference.

Also shown in the figure are the dependencies among the components and subsystems. Roughly speaking, the Event Coreference subsystem, which lies at the end of the information extraction pipeline, relies on the outputs of the Event Extraction subsystem and the Entity Coreference subsystem, which in turn rely on the output of the Entity Extraction subsystem. Below we describe the components in each subsystem in detail.

2.1. Entity Extraction Subsystem

The Entity Extraction subsystem consists of three components, the Entity Mention Identification component (Component 1), the Entity Typing and Subtyping component (Component 2), and the Named Entity Recognition component (Component 3). The outputs of these components will subsequently be used by the Entity Coreference subsystem and the Event Extraction subsystem, so they only have an indirect influence on event coreference.

Component 1: Entity Mention Identification

This component extracts from raw text the entity mentions and the candidate event arguments. Event arguments can be entity mentions, time expressions, and value expressions.[3]

We recast the task of identifying entity mentions, time expressions, and value expressions as a sequence labeling task, where we train one CRF (using the CRF++ software

[1] SinoCoreferencer can be downloaded from `http://www.hlt.utdallas.edu/~yzcchen/coref`.

[2] `http://nlp.stanford.edu/software/corenlp.shtml`

[3] The ACE 2005 task definition is available from http://`http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v2a.pdf`.

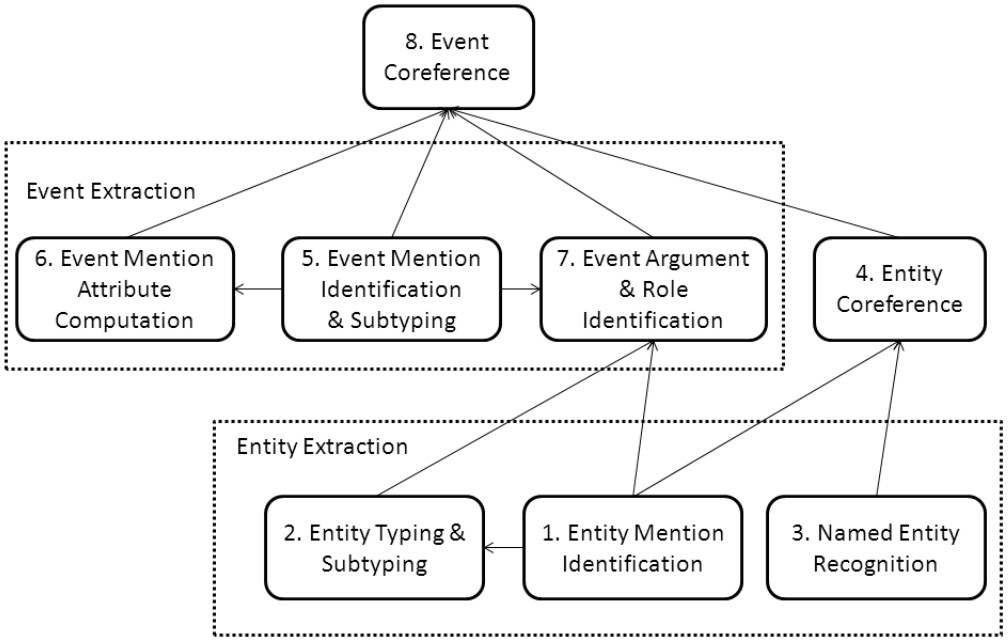

Figure 1: The architecture of SinoCoreferencer.

package[4]) to extract each of these three types of candidate event arguments. Specifically, we create one instance for each character c_i, assigning it a class label that indicates whether it begins a candidate event argument, is inside an argument or is outside an argument. So there are three class labels in total. Below we describe the 25 features we use to represent c_i, which can be divided into three categories: lexical, wordlist-based, and grammatical. The number enclosed within the parentheses after each category is the number of features in that category.

Lexical (12): Character unigrams (c_{i-3}, c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2}, c_{i+3}), character bigrams ($c_{i-1}c_i$, c_ic_{i+1}), and character trigrams ($c_{i-2}c_{i-1}c_i$, $c_{i-1}c_ic_{i+1}$, $c_ic_{i+1}c_{i+2}$) formed from characters in a window of five.

Wordlist-based (10): We employ 10 Chinese wordlists to generate 10 features, each of which is computed based on exactly one wordlist. The 10 wordlists consist of (a) Chinese surnames; (b) famous GPE[5] and location names (three wordlists); (c) Chinese location suffixes; (d) Chinese GPE suffixes; (e) famous international organization names; (f) famous company names; (g) famous person names; and (h) a list of pronouns. To create the wordlist-based features, we first create $n+1$ strings, each of which is of the form $s_j = c_i...c_{i+j}$, where $0 \leq j \leq n$ and n is the maximum length of a string that can be found in the 10 wordlists. Then we check whether any of these strings is in any of these wordlists. The default feature value for all 10 wordlists is O. If s_i appears in wordlist k, we (1) set $k(c_i)$, the feature corresponding to wordlist k for c_i, to B, and (2) set $k(x)$ to I, where x is a character in s_i that is not c_i.

Grammatical (3): The part-of-speech tag of c_i concatenated with either "B-" or "I-" to indicate whether c_i is the first character of c_i or not; whether c_i is in a NP or not; whether c_i is part of a pronoun.

[4] http://code.google.com/p/crfpp/

[5] GPE (geo-political entity) is an entity type defined in ACE. Examples of GPEs include country and city names.

Type	SubTypes
Facility	Airport, Building-Grounds, Path, Plant, Subarea-Facility
Geo-Political Entity	Continent, County-or-District, GPE-Cluster, Nation, Population-Center, Special, State-or-Province
Location	Address, Boundary, Celestial, Land-Region-Natural, Region-General, Region-International, Water-Body
Organization	Commercial, Educational, Entertainment, Government, Media, Medical-Science, Non-Governmental, Religious, Sports
Person	Group, Indeterminate, Individual
Vehicle	Air, Land, Subarea-Vehicle, Underspecified, Water
Weapon	Biological, Blunt, Chemical, Exploding, Nuclear, Projectile, Sharp, Shooting, Underspecified

Table 1: ACE 2005 entity types and subtypes.

Component 2: Entity Typing and Subtyping

This component takes a set of entity mentions (provided by Component 1) and determines the semantic type and subtype of each of them. Since we train this component on the ACE 2005 training data, the entity types and subtypes it produces are those defined in the ACE 2005 task definition. The complete list of ACE 2005 entity types and subtypes is shown in Table 1. As we can see, there are seven entity types and 45 entity subtypes.

Knowing the semantic type and subtype of an argument is

helpful for classifying the role of event arguments. For example, we can assign the role VICTIM only to those arguments with entity type PERSON. To determine semantic types and subtypes, we train two SVM multiclass classifiers using SVM^{multiclass} (Tsochantaridis et al., 2004). We create one training instance for each mention m_k. Its class label is either the semantic type or the semantic subtype of m_k, depending on which classifier we are training. We employ the same set of features for representing an instance when training the two classifiers, as described below.

Lexical (6): m_k's head string; each character in m_k's head; characters in a window of five surrounding m_k's head.

Wordlist-based (10): We employ the same 10 wordlists that we used in the entity mention identification task to generate 10 features. Those 10 features indicate whether m_k appears in these 10 wordlists.

Semantic (1): The semantic category of m_k's head, which is extracted from a Chinese lexical database organized in a similar way as the English WordNet.[6]

Component 3: Named Entity Recognition

We train our named entity tagger using the CRF++ software package on the Chinese training data of the CoNLL-2012 shared task. The CoNLL-2012 data is annotated with 18 named entity types, as listed in the leftmost column of Table 4.[7]

We recast named entity recognition as a sequence labeling task. Specifically, we create one instance for each character c_i, assigning c_i a class label that indicates whether it begins a specific named entity (one of 18 labels), is inside a specific named entity (one of 18 labels) or is outside a named entity (1 label). So there are 37 class labels in total. Below we describe the 18 features, which can be divided into three categories: lexical, wordlist-based, and grammatical.

Lexical (12): Character unigrams (c_{i-3}, c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2}, c_{i+3}), character bigrams ($c_{i-1}c_i$, c_ic_{i+1}), and character trigrams ($c_{i-2}c_{i-1}c_i$, $c_{i-1}c_ic_{i+1}$, $c_ic_{i+1}c_{i+2}$) formed from characters in a window of seven.

Wordlist-based (3): Whether c_i is in a Chinese surname wordlist; whether c_i is in a Chinese location suffix wordlist; whether c_i is in a Chinese GPE suffix wordlist.

Grammatical (3): The part-of-speech tag of c_i concatenated with either "B-" or "I-" to indicate whether c_i is the first character of c_i or not; whether c_i is in a NP or not; whether c_i is inside part of a pronoun.

2.2. Entity Coreference Subsystem

The Entity Coreference subsystem has only one component (Component 4).

[6]The dictionary is available from Harbin Institute of Technology's NLP Group website.

[7]The reason why we train our named entity recognizer on the CoNLL-2012 shared task data rather than the ACE 2005 data is that the former contains more named entity annotations than the latter.

Component 4: Entity Coreference

The Entity Coreference component creates a coreference partition in which each cluster contains all and only those entity mentions that refer to the same real-world entity. Since two event mentions having coreferent arguments are likely to be coreferent, the output of this component can be used to create useful features for event coreference.

This component assumes as input a set of entity mentions (identified by Component 1) for a document, and then generates features for these mentions from the syntactic parse trees (provided by the Stanford CoreNLP tool) and the named entities (provided by Component 3). The coreference resolution algorithm used by this component adopts a sieve-based approach, as described in detail below.

The sieve-based approach to coreference resolution was originally proposed by Raghunathan et al. (2010). Informally, *sieve* is composed of one or more heuristic *rules*. Each rule extracts a coreference relation between two mentions based on one or more *conditions*. Sieves are ordered by their precision, with the most precise sieve appearing first. To resolve a set of mentions in a document, the resolver makes multiple passes over them: in the *i*-th pass, it attempts to use only the rules in the *i*-th sieve to find an antecedent for each mention m_k. Specifically, when searching for an antecedent for m_k, its candidate antecedents are visited in an order determined by their positions in the associated parse tree (Haghighi and Klein, 2009). The partial clustering of the mentions created in the *i*-th pass is then passed to the *i*+1-th pass. Hence, later passes can exploit the information computed by previous passes, but a coreference link established earlier cannot be overridden later.

For Chinese coreference resolution, we design nine sieves, some of which are motivated by the sieves proposed by Lee et al. (2011) for English coreference resolution. Below we describe each of these sieves in detail.

1. **Discourse Processing sieve:** There are four scenarios in which this sieve posits two mentions as coreferent.
   - Two mentions are both first person pronouns and have the same speaker.
   - One mention is a first person pronoun in a dialogue, of which the other mention is the speaker.
   - Two mentions are both second person pronouns in two dialogues with the same speaker.
   - One mention is a first person pronoun and the other is a second person pronoun. Also, the speaker of the first person pronoun should be the same as the subsequent analogue's speaker of the second person pronoun.
2. **Exact Match sieve:** This sieve posits two mentions having exactly the same string as coreferent if they are not pronouns.
3. **Precise Constructs sieve:** This sieve employs named entity information. It handles specific cases of abbreviations for Chinese named entities: (a) Abbreviation of foreign person names, e.g., *萨达姆·侯赛因 [Saddam Hussein] and 萨达姆 [Saddam]*. (b) Abbreviation of Chinese person names, e.g., *陈总统 [Chen*

President] and 陈水扁总统 [Chen Shui-bian President]. (c) Abbreviation of country names, e.g, *多国 [Do country] and 多米尼加 [Dominica]*.

4. **Strict Head Match sieve A:** A mention m_k and a candidate antecedent m_j are posited as coreferent if they satisfy all of the following conditions: (a) their head nouns are the same; (b) all the non-stop words in m_k appears in at least one of the mentions in the same cluster as m_j; (c) all the modifiers in m_k appear in m_j, including includes all adjective and nouns modifiers; and (d) m_k and m_j are not in an i-within-i construct (see Raghunathan et al. (2010)).

5. **Strict Head Match sieve B:** A relaxed version of the Strict Head Match sieve A that posits two mentions as coreferent as long as the aforementioned conditions (a), (b) and (d) are satisfied.

6. **Strict Head Match sieve C:** Another relaxed version of the Strict Head Match sieve A that posits two mentions as coreferent as long as the aforementioned conditions (a), (c) and (d) are satisfied.

7. **Proper Head Word Match sieve:** This sieve relaxes Strict Head Match sieve A by deleting conditions (b) and (c), but it requires that the two mentions' head nouns are proper nouns and the two mentions cannot have different location or number modifiers.

8. **Pronouns sieve:** The Pronouns sieve resolves a pronoun to the closest preceding mention whose *gender*, *number* and *person* are compatible with those of the pronoun.

9. **Lexical Pair sieve:** This sieve exploits lexical information using a learning-based approach. Given a pair of mentions m_j and m_k in the test data, two probabilities are computed based on the ACE training data: (1) SP-Prob, the *string-pair* probability, which is the probability that two strings of two mentions are coreferent in the training data; and (2) HP-Prob, the *head-pair* probability, which is the probability that two heads of two mentions are coreferent in the training data. These two probabilities affect the whole system in two aspects. We set two thresholds, t_{SPL} and t_{HPL}. If SP-Prob$\leq t_{SPL}$ or HP-Prob$\leq t_{HPL}$, even when the pair of m_j and m_k satisfies the conditions specified in any of the above sieves, our resolver will not posit them as coreferent. We set two other thresholds, t_{SPU} and t_{HPU}. If SP-Prob$\geq t_{SPU}$ or HP-Prob$\geq t_{HPU}$, our resolver will posit them as coreferent no matter what. These four thresholds are tuned on development data.

2.3. Event Extraction Subsystem

The Event Extraction subsystem consists of three components, the Event Mention Identification and Subtyping component (Component 5), the Event Mention Attribute Computation component (Component 6), and the Event Argument and Role Identification component (Component 7).

Type	SubTypes
Life	Be-Born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-Money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End-Org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-Write
Personnel	Start-Position, End-Position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Table 2: ACE 2005 event types and subtypes.

Component 5: Event Mention Identification and Subtyping
This component (1) provides the event mentions for event coreference resolution, and (2) labels each event mention with its subtype. Since we train this component on the ACE 2005 training data, the event subtypes it produces are those that are defined in the ACE 2005 annotation guidelines. The complete list of event subtypes is shown in Table 2. As we can see, there are 33 event subtypes, which can be categorized into eight broader event types.
Since two event mentions with different subtypes cannot be coreferent, subtypes can be used to create useful features for event coreference. To implement this component, we use our Chinese event extraction system (Chen and Ng, 2012), which jointly learns these tasks via training a classifier using the SVM^{light} software package.

Component 6: Event Mention Attribute Value Computation
This component takes as input a set of event mentions (provided by Component 5) and computes for each mention its attributes, including its POLARITY, MODALITY, GENERICITY and TENSE. Since two event mentions that differ with respect to any of these attributes cannot be coreferent, they can be used to create useful features for event coreference. Following Chen et al. (2009), we employ a classifier trained on the ACE 2005 training data using maximum entropy modeling[8] to compute the value of each attribute of each event mention (see Chen et al. for details on the implementation of these classifiers, including the features used to train each classifier).

Component 7: Event Argument and Role Classification
This component takes as input a set of event mentions (provided by Component 5) and a set of candidate event arguments (provided by Component 1). For each event mention em, it (1) identifies those candidate arguments that are the true arguments of em (e.g., the participants, time, and place

[8] We use the maximum entropy implementation available at `http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html`.

Input	Entity Typing			Entity Subtyping		
	R	P	F	R	P	F
Perfect	90.1	90.1	90.1	81.6	81.6	81.6
Predicted	80.5	77.6	79.0	73.1	70.4	71.7

Table 3: Entity typing and subtyping performance.

Type	R	P	F
CARDINAL	55.8	54.3	55.0
DATE	76.8	77.1	76.9
EVENT	18.4	61.0	28.3
FAC	21.0	69.6	32.2
GPE	85.0	66.6	74.7
LANGUAGE	44.4	40.0	42.1
LAW	16.7	76.9	27.4
LOC	27.7	68.2	39.4
MONEY	91.9	75.6	82.9
NORP	28.8	66.7	40.2
ORDINAL	77.9	85.1	81.3
ORG	69.9	54.6	61.3
PERCENT	83.1	85.2	84.2
PERSON	77.5	69.9	73.5
PRODUCT	2.0	25.0	3.8
QUANTITY	61.5	79.1	69.2
TIME	62.1	85.7	72.0
WORK_OF_ART	19.5	60.4	29.4
Overall	62.9	70.3	66.4

Table 4: Named entity recognition performance.

of em), and then (2) assigns a *role* (e.g., VICTIM, PLACE, TIME-WITHIN) to each of its true arguments. Since two events involving different times, places, or participants cannot be coreferent, the arguments and their roles can be used to create useful features for event coreference. To implement this component, we use our Chinese event extraction system (Chen and Ng, 2012), which jointly learns these two tasks by training a classifier using the SVM^{light} software package.

2.4. Event Coreference Subsystem

The Event Coreference subsystem has only one component (Component 8).

Component 8: Event Coreference

Underlying our learning-based event coreference resolver is a mention-pair model (Soon et al., 2001) trained using the SVM^{light} software package. Training instances are created as follows. For each anaphoric event mention em, we create one positive instance between em and its closest antecedent. To create negative instances, we pair em with each of its preceding event mentions that is not coreferent with em.

Each instance is represented using 32 features, which are modeled after a state-of-the-art English event coreference resolver (Chen and Ji, 2009; Chen et al., 2009). The 32 features can be divided into five groups, as discussed below. For convenience, we use em_2 to refer to an event mention to be resolved and em_1 to refer to one of its candidate antecedents.

Group 1 (Event Type and Subtype features). The four features in this group are provided by Component 5. These features encode: whether em_1 and em_2 agree w.r.t. event type; whether they agree w.r.t. event subtype; the concatenation of their event types; and the concatenation of their event subtypes.

Group 2 (Event Mention Attribute features). The eight features in this group are computed based on the output of Component 6. These features encode: the four event mention attributes of em_2; and whether em_1 and em_2 are compatible w.r.t. each of the event mention attributes.

Group 3 (Event Argument Roles features). The four features in this group are computed based on the output of Component 7. These features encode: the roles and number of the arguments that only appear in em_1; and the roles and number of the arguments that only appear in em_2.

Group 4 (Entity Coreference features). The six features in this group are computed based on the output of Component 4. These features encode: the roles and number of arguments between em_1 and em_2 that have the same role and are also in the same entity coreference chain; the roles and number of arguments between em_1 and em_2 that have same role but are in different coreference chains; and the roles and number of arguments between em_1 and em_2 that have different roles but are in the same coreference chain.

Group 5 (Other features). The 10 features in this group encode: the sentence distance between em_1 and em_2; the number of event mentions intervening them in the associated text; the number of words between them; whether they have the same trigger word; whether they are in a coordinating structure; whether they have same basic verb[9]; whether they agree in number if they are nouns; whether they have incompatible modifiers if they are nouns; the concatenation of the part-of-speech tags of their heads; and the concatenation of their trigger words.

After training, the resulting mention-pair model is used in combination with a closest-first single-link clustering algorithm to impose a coreference partition on the event mentions in a test text (Soon et al., 2001). The test instances are created in the same way as the training instances.

3. Evaluation

Despite the fact that the focus of this paper is event coreference, we will evaluate each of the eight components of SinoCoreferencer in this section. As we mentioned in the introduction, each of its components can be used in a standalone manner. Presenting the results of each component will therefore give an end user of a particular component a better idea of whether it is accurate enough to be used in her application.

3.1. Experimental Setup

Corpus. Evaluations of all but the Named Entity Recognition component will be performed on the Chinese portion of the ACE 2005 training corpus[10] using 5-fold cross validation. The ACE 2005 training corpus consists of 633 documents with 3,333 event mentions distributed over 2,521

[9] For example, the basic verb of 回家 (go home) is 回 (go).

[10] The ACE 2005 test documents are not publicly available.

	MUC			B^3			$CEAF_e$			Avg
Input	R	P	F	R	P	F	R	P	F	F
Perfect	71.5	85.8	78.0	67.4	88.0	76.4	69.4	48.8	57.3	70.6
Predicted	61.7	78.0	68.9	63.6	84.6	72.6	57.9	40.3	47.6	63.0

Table 5: Entity coreference performance.

	POLARITY			MODALITY			GENERICITY			TENSE		
Input	R	P	F	R	P	F	R	P	F	R	P	F
Perfect	96.5	96.5	96.5	86.9	86.9	86.9	91.2	91.2	91.2	67.1	67.1	67.1
Predicted	57.9	68.8	62.9	51.8	61.6	56.2	54.7	65.0	59.4	33.8	40.2	36.7

Table 6: Event attribute classification performance.

	MUC			B^3			$CEAF_e$			Avg
Input	R	P	F	R	P	F	R	P	F	F
Perfect	80.4	70.0	74.8	88.4	79.7	83.8	57.3	66.8	61.7	73.4
Predicted	37.4	36.7	37.1	72.8	71.1	71.9	40.6	41.1	40.8	49.9

Table 7: Event coreference performance.

	Argument			Role		
Input	R	P	F	R	P	F
Perfect	68.9	87.1	76.9	61.1	77.2	68.2
Predicted	23.1	36.7	28.3	20.0	31.9	24.6

Table 8: Argument identification and role classification performance.

event coreference chains and 34,319 entity mentions distributed over 15,413 entity coreference chains.

The performance of the named entity recognizer will be measured on the CoNLL-2012 shared task corpus. Specifically, we train the recognizer on the CoNLL-2012 training set and evaluate its performance on the CoNLL-2012 development set.[11] The CoNLL-2012 training set contains 1,391 documents with 62,543 named entities, while the development set contains 172 documents with 9,104 named entities.

Evaluation metrics. To evaluate entity and event coreference performance, we employ three commonly-used coreference scoring measures, namely MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_e$ (Luo, 2005). Each of these evaluation measures reports results in terms of recall (R), precision (P), and F-score (F). In addition, following the CoNLL-2012 shared task on unrestricted coreference, we also report the unweighted average (Avg) of the F-scores produced by these three metrics. To evaluate the remaining components, we employ recall, precision, and F-score computed in the standard manner.

Evaluation settings. We evaluate each component under two settings, which differ in terms of whether the component has access to perfect or predicted input. For example, the input to the Entity Coreference component comes from the Entity Mention Identification component and the Named Entity Recognition component, so evaluating the Entity Coreference component under the perfect (predicted) setting implies that it is given access to perfect (predicted) information regarding entity mentions and named entities. Comparing the results of these two settings enables us to get a better idea of how much performance deterioration can be attributed to noisy input.

[11] We do not use the CoNLL-2012 test set for evaluating the named entity recognizer because the version we have does not contain named entity annotations.

3.2. Results

Next, we present the results of each of the eight components.

Component 1: Entity Mention Identification. Recall that this component does not depend on any other components. Its recall, precision, and F-score are 86.8%, 82.6% and 84.7% respectively.

Component 2: Entity Typing and Subtyping. Results of this component, which depends on Component 1, are shown in Table 3. Given perfect input, its entity typing and subtyping F-scores are 90.1% and 81.6% respectively. In contrast, given predicted input, its entity typing and subtyping F-scores are 79.0% and 71.7% respectively.

Component 3: Named Entity Recognition. Recall that this component does not depend on any other components. Its overall and per-class results are shown in Table 4. As we can see, it achieves an overall F-score of 66.4%. Note that reasonably high F-scores are achieved for PERSON, GPE, ORGANIZATION,PERCENT, MONEY, QUANTITY and CARDINAL.

Component 4: Entity Coreference. Results of this component, which depends on Components 1 and 3, are shown in Table 5. Given perfect input, it achieves an Avg F-score of 70.6%. In contrast, given predicted input, it achieves an Avg F-score of 63.0%.

Component 5: Event Mention Identification and Subtyping. This component does not dependent on any other components. Our event mention identifier achieves scores of 60.0% (R), 71.3% (P) and 65.1% (F), whereas our event subtype classifier achieves scores of 56.4% (R), 67.1% (P) and 61.30% (F).

Component 6: Event Mention Attribute Value Computation. Recall that we trained four classifiers to predict

the four attribute values of an event mention. Results of these four classifiers are shown in Table 6. Given gold event mentions, the POLARITY, MODALITY, GENERICITY, and TENSE classifiers achieve F-scores of 96.5%, 86.9%, 91.2%, and 67.1% respectively. In contrast, given predicted event mentions, the F-scores achieved by these four classifiers are 62.9%, 56.2%, 59.4%, and 36.7% respectively.

Component 7: Event Argument and Role Classification. Results of this component, which depends on Components 1, 2, and 5, are shown in Table 8. Given perfect input, the F-scores for argument identification and role classification are 76.9% and 68.2% respectively. Given predicted input, the corresponding F-scores drop to 28.3% and 24.6%.

Component 8: Event Coreference. Results of this component, which depends on Components 4, 5, 6, and 7, are shown in Table 7. Given perfect input, the Avg F-score is 73.4%. In contrast, given predicted input, the Avg F-score drops to 49.9%.

4. Conclusion

We described the design, implementation, and evaluation of SinoCoreferencer, a publicly-available end-to-end ACE-style Chinese event coreference system that achieved state-of-the-art performance on the ACE 2005 corpus. To our knowledge, SinoCoreferencer is the first publicly-available Chinese event coreference resolver. We hope that its underlying components can be profitably exploited to develop high-level natural language applications for Chinese.

5. Acknowledgments

We thank the three reviewers for their insightful comments. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of NSF.

6. References

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation*, page 563--566.

Bejan, C. and Harabagiu, S. (2010). Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412--1422.

Chen, Z. and Ji, H. (2009). Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 54--57.

Chen, C. and Ng, V. (2012). Joint modeling for Chinese event extraction with rich linguistic features. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 529--544.

Chen, Z., Ji, H., and Haralick, R. (2009). A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the RANLP 2009 Workshop on Events in Emerging Text Types*, pages 17--22.

Chen, B., Su, J. and Tan, C. L. (2010). Resolving Event Noun Phrases to Their Verbal Mentions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 872--881.

Chen, B., Su, J., Pan, S. J., and Tan, C. L. (2011). A unified event coreference resolution by integrating multiple resolvers. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 102--110.

Haghighi, A. and Klein, D. (2009). Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152--1161.

Humphreys, K., Gaizauskas, R., and Azzam, S. (1997). Event coreference for information extraction. In *Proceedings of the Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 75--81.

Joachims, T. (1999). Making large-scale SVM learning practical. In Scholkopf, B. and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*, pages 44--56. MIT Press.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28--34.

Lee, H., Recasens, M., Chang, A., Surdeanu, M., and Jurafsky, D. (2012). Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489--500.

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.

Mitkov, R., Boguraev, B., and Lappin, S. (2001). Introduction to the special issue on computational anaphora resolution. *Computational Linguistics*, 27(4):473--477.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., and Manning, C. (2010). A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492--501.

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.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning*, pages 104--112.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference*, pages 45--52.