The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies

Mihai Surdeanu†,* **Richard Johansson**‡ **Adam Meyers**$^{\diamond}$
Lluís Màrquez†† **Joakim Nivre**‡‡,**

$\dagger$: Barcelona Media Innovation Center, mihai.surdeanu@barcelonamedia.org
$\star$: Yahoo! Research Barcelona, mihais@yahoo-inc.com
$\ddagger$: Lund University, richard@cs.lth.se
$\diamond$: New York University, meyers@cs.nyu.edu
$\dagger\dagger$: Technical University of Catalonia, lluism@lsi.upc.edu
$\ddagger\ddagger$: Växjö University, joakim.nivre@vxu.se
$\star\star$: Uppsala University, joakim.nivre@lingfil.uu.se

Abstract

The Conference on Computational Natural Language Learning is accompanied every year by a shared task whose purpose is to promote natural language processing applications and evaluate them in a standard setting. In 2008 the shared task was dedicated to the joint parsing of syntactic and semantic dependencies. This shared task not only unifies the shared tasks of the previous four years under a unique dependency-based formalism, but also extends them significantly: this year's syntactic dependencies include more information such as named-entity boundaries; the semantic dependencies model roles of both verbal and nominal predicates. In this paper, we define the shared task and describe how the data sets were created. Furthermore, we report and analyze the results and describe the approaches of the participating systems.

1 Introduction

In 2004 and 2005 the shared tasks of the Conference on Computational Natural Language Learning (CoNLL) were dedicated to semantic role labeling (SRL), in a monolingual setting (English). In 2006 and 2007 the shared tasks were devoted to the parsing of syntactic dependencies, using corpora from up to 13 languages. The CoNLL-2008 shared task[1] proposes a unified dependency-based formalism, which models both syntactic dependencies and semantic roles. Using this formalism, this shared task merges both the task of syntactic dependency parsing and the task of identifying semantic arguments and labeling them with semantic roles. Conceptually, the 2008 shared task can be divided into three subtasks: (i) parsing of syntactic dependencies, (ii) identification and disambiguation of semantic predicates, and (iii) identification of arguments and assignment of semantic roles for each predicate. Several objectives were addressed in this shared task:

- SRL is performed and evaluated using a dependency-based representation for both syntactic and semantic dependencies. While SRL on top of a dependency treebank has been addressed before (Hacioglu, 2004), our approach has several novelties: (i) our constituent-to-dependency conversion strategy transforms all annotated semantic arguments in PropBank and NomBank not just a subset; (ii) we address propositions centered around both verbal (PropBank) and nominal (NomBank) predicates.
- Based on the observation that a richer set of syntactic dependencies improves semantic processing (Johansson and Nugues, 2007), the syntactic dependencies modeled are more complex than the ones used in the previous CoNLL shared tasks. For example, we now include apposition links, dependencies derived from named entity (NE) structures, and better modeling of long-distance grammatical relations.
- A practical framework is provided for the joint learning of syntactic and semantic dependencies.

[1] `http://www.yr-bcn.es/conll2008`

Given the complexity of this shared task, we limited the evaluation to a monolingual, English-only setting. The evaluation is separated into two different challenges: a closed challenge, where systems have to be trained strictly with information contained in the given training corpus, and an open challenge, where systems can be developed making use of any kind of external tools and resources. The participants could submit results in either one or both challenges.

This paper is organized as follows. Section 2 defines the task, including the format of the data, the evaluation metrics, and the two challenges. Section 3 introduces the corpora used and our constituent-to-dependency conversion procedure. Section 4 summarizes the results of the submitted systems. Section 5 discusses the approaches implemented by participants. Section 6 analyzes the results using additional non-official evaluation measures. Section 7 concludes the paper.

2 Task Definition

In this section we provide the definition of the shared task, starting with the format of the shared task data, followed by a description of the evaluation metrics used and a discussion of the two shared task challenges, i.e., closed and open.

2.1 Data Format

The data format used in this shared task was highly influenced by the formats used in the 2004–2007 shared tasks. The data follows these general rules:

- The files contain sentences separated by a blank line.
- A sentence consists of one or more tokens and the information for each token is represented on a separate line.
- A token consists of at least 11 fields. The fields are separated by one or more whitespace characters (spaces or tabs). Whitespace characters are not allowed within fields.

Table 1 describes the fields stored for each token in the closed-track data sets. Columns 1–3 and 5–8 are available at both training and test time. Column 4, which contains gold-standard part-of-speech (POS) tags, is not given at test time. The same holds for columns 9 and above, which contain the syntactic and semantic dependency structures that the systems should predict.

The PPOS and PPOSS fields were automatically predicted using the SVMTool POS tagger (Giménez, 2004). To predict the tags in the training set, a 5-fold cross-validation procedure was used. The LEMMA and SPLIT_LEMMA fields were predicted using the built-in lemmatizer in WordNet (Fellbaum, 1998) based on the most frequent sense for the form and part-of-speech tag.

Since NomBank uses a sub-word analysis in some hyphenated words (such as $[\textit{finger}]_{\mathrm{ARG}}$-$[\textit{pointing}]_{\mathrm{PRED}}$), the data format represents the parts in hyphenated words as separate tokens (columns 6–8). However, the format also represents how the parts originally fit together before splitting (columns 2–5). Padding characters ("_") are used in columns 2–5 to ensure the same number of rows for all columns corresponding to one sentence. All syntactic and semantic dependencies are annotated relative to the split word forms (columns 6–8).

Table 2 shows the columns available to the systems participating in the open challenge: named-entity labels as in the CoNLL-2003 Shared Task (Tjong Kim San and De Meulder, 2003) and from the BBN Wall Street Journal Entity Corpus,[2] WordNet supersense tags, and the output of an off-the-shelf dependency parser (Nivre et al., 2007b). Columns 1–3 were predicted using the tagger of Ciaramita and Altun (2006). Because the BBN corpus shares lexical content with the Penn Treebank, we generated the BBN tags using a 2-fold cross-validation procedure.

2.2 Evaluation Measures

We separate the evaluation measures into two groups: (i) official measures, which were used for the ranking of participating systems, and (ii) additional unofficial measures, which provide further insight into the performance of the participating systems.

2.2.1 Official Evaluation Measures

The official evaluation measures consist of three different scores: (i) syntactic dependencies are scored using the labeled attachment score (LAS), (ii) semantic dependencies are evaluated using a labeled F_1 score, and (iii) the overall task is scored with a macro average of the two previous scores. We describe all these scoring measures next.

The LAS score is defined similarly as in the previous two shared tasks, as the percentage of to-

[2] LDC catalog number LDC2005T33.

Number	Name	Description
1	ID	Token counter, starting at 1 for each new sentence.
2	FORM	Unsplit word form or punctuation symbol.
3	LEMMA	Predicted lemma of FORM.
4	GPOS	Gold part-of-speech tag from the Treebank (empty at test time).
5	PPOS	Predicted POS tag.
6	SPLIT_FORM	Tokens split at hyphens and slashes.
7	SPLIT_LEMMA	Predicted lemma of SPLIT_FORM.
8	PPOSS	Predicted POS tags of the split forms.
9	HEAD	Syntactic head of the current token, which is either a value of ID or zero (0).
10	DEPREL	Syntactic dependency relation to the HEAD.
11	PRED	Rolesets of the semantic predicates in this sentence.
12...	ARG	Columns with argument labels for each semantic predicate following textual order.

Table 1: Column format in the closed-track data. The columns in the lower part of the table are unseen at test time and are to be predicted by systems.

Number	Name	Description
1	CONLL2003	Named entity labels using the tag set from the CoNLL-2003 shared task.
2	BBN	NE labels using the tag set from the BBN Wall Street Journal Entity Corpus.
3	WNSS	WordNet super senses.
4	MALT_HEAD	Head of the syntactic dependencies generated by MaltParser.
5	MALT_DEPREL	Label of syntactic dependencies generated by MaltParser.

Table 2: Column format in the open-track data.

kens for which a system has predicted the correct HEAD and DEPREL columns (see Table 1). Same as before, our scorer also computes the unlabeled attachment score (UAS), i.e., the percentage of tokens with correct HEAD, and label accuracy, i.e., the percentage of tokens with correct DEPREL.

The semantic propositions are evaluated by converting them to semantic dependencies, i.e., we create a semantic dependency from every predicate to all its individual arguments. These dependencies are labeled with the labels of the corresponding arguments. Additionally, we create a semantic dependency from each predicate to a virtual ROOT node. The latter dependencies are labeled with the predicate senses. This approach guarantees that the semantic dependency structure conceptually forms a single-rooted, connected (but not necessarily acyclic) graph. More importantly, this scoring strategy implies that if a system assigns the incorrect predicate sense, it still receives some points for the arguments correctly assigned. For example, for the correct proposition:

```
verb.01: ARG0, ARG1, ARGM-TMP
```

the system that generates the following output for the same argument tokens:

```
verb.02: ARG0, ARG1, ARGM-LOC
```

receives a labeled precision score of 2/4 because two out of four semantic dependencies are incorrect: the dependency to ROOT is labeled `02` instead of `01` and the dependency to the `ARGM-TMP` is incorrectly labeled `ARGM-LOC`. Using this strategy we compute precision, recall, and F_1 scores for both labeled and unlabeled semantic dependencies.

Finally, we combine the syntactic and semantic measures into one global measure using macro averaging. We compute macro precision and recall scores by averaging the labeled precision and recall for semantic dependencies with the LAS for syntactic dependencies:[3]

$$LMP = W_{sem} * LP_{sem} + (1 - W_{sem}) * LAS \qquad (1)$$

$$LMR = W_{sem} * LR_{sem} + (1 - W_{sem}) * LAS \qquad (2)$$

where LMP is the labeled macro precision and LP_{sem} is the labeled precision for semantic dependencies. Similarly, LMR is the labeled macro recall and LR_{sem} is the labeled recall for semantic dependencies. W_{sem} is the weight assigned to the semantic task.[4] The macro labeled F_1 score, which was used for the ranking of the participating systems, is computed as the harmonic mean of LMP and LMR.

[3] We can do this because the LAS for syntactic dependencies is a special case of precision and recall, where the predicted number of dependencies is equal to the number of gold dependencies.

[4] We assign equal weight to the two tasks, i.e., $W_{sem} = 0.5$.

2.2.2 Additional Evaluation Measures

We used several additional evaluation measures to further analyze the performance of the participating systems.

The first additional measure used is *Exact Match*, which reports the percentage of sentences that are completely correct, i.e., all the generated syntactic dependencies are correct and all the semantic propositions are present and correct. While this score is significantly lower than any of the official scores, it will award systems that performed joint learning or optimization for all subtasks.

In the same spirit but focusing on the semantic subtasks, we report the *Perfect Proposition* F_1 score, where we score entire semantic frames or propositions. This measure is similar to the PProps accuracy score from the 2005 shared task (Carreras and Màrquez, 2005), with the caveat that this year this score is implemented as an F_1 measure, because predicates are not provided in the test data. Hence, propositions may be over or under generated at prediction time.

Lastly, we analyze systems based on the ratio between labeled F_1 score for semantic dependencies and the LAS for syntactic dependencies. In other words, this measure normalizes the semantic scores relative to the performance of the parsing component. This measure estimates the true overall performance of the semantic subtasks, independent of the syntactic parser.[5] For example, this score addresses the situations where the semantic labeled F_1 score of one system is artificially low because the corresponding syntactic component does not perform well.

2.3 Closed and Open Challenges

Similarly to the CoNLL-2005 shared task, this shared task evaluation is separated into two challenges:

Closed Challenge - systems have to be built strictly with information contained in the given training corpus, and tuned with the development section. In addition, the PropBank and NomBank lexical frames can be used. These restrictions mean that constituent-based parsers or SRL systems can not be used in this challenge because the constituent-based annotations are not provided in our training set. The aim of this challenge is to compare the performance of the participating systems in a fair environment.

Open Challenge - systems can be developed making use of any kind of external tools and resources. The only condition is that such tools or resources must not have been developed with the annotations of the test set, both for the input and output annotations of the data. In this challenge, we are interested in learning methods which make use of any tools or resources that might improve the performance. For example, we encourage the use of semantic information, as provided by NE recognition or word-sense disambiguation (WSD) systems (such state-of-the-art annotations are provided by the organizers, see Table 2). Also, in this challenge participants are encouraged to use constituent-based parsers and SRL systems, as long as these systems were trained only with the sections of Penn Treebank used in the shared task training corpus. To encourage the participation of the groups that are only interested in SRL, the organizers provide also the output of a state-of-the-art dependency parser as input in this challenge. The comparison of different systems in this setting may not be fair, and thus ranking of systems is not necessarily important.

3 Data

The corpora used in the shared task evaluation were generated through a process that merges several input corpora and converts them from the constituent-based formalism to dependencies. This section starts with an introduction of the input corpora used, followed by a description of the constituent-to-dependency conversion process. The section concludes with an overview of the shared task corpora.

3.1 Input Corpora

Input to our merging procedures includes the Penn Treebank, BBN's named entity corpus, PropBank and NomBank. In this section, we will provide brief descriptions of these annotations in terms of both form and content. All annotations are currently being distributed by the Linguistic Data Consortium, with the exception of NomBank, which is freely downloadable.[6]

[5] A correct evaluation of the stand-alone SRL systems would require the usage of gold syntactic dependencies, but these were not provided for the testing corpora.

[6] `http://nlp.cs.nyu.edu/meyers/NomBank.html`

3.1.1 Penn Treebank 3

The Penn Treebank 3 corpus (Marcus et al., 1993) consists of hand-coded parses of the Wall Street Journal (test, development and training) and a small subset of the Brown corpus (W. N. Francis and H. Kuĉera, 1964) (test only). These hand parses are notated in-line and sometimes involve changing the strings of the input data. For example, in file wsj_0309, the token *fearlast* in the text corresponds to the two tokens *fear* and *last* in the annotated data. In a similar way, *cannot* is regularly split to *can* and *not*. It is significant that the other annotations assume the tokenization of the Penn Treebank, as this makes it easier for us to merge the annotation. The Penn Treebank syntactic annotation includes phrases, parts of speech, empty category representations of various filler/gap constructions and other phenomena, based on a theoretical perspective similar to that of Government and Binding Theory (Chomsky, 1981).

3.1.2 BBN Pronoun Coreference and Entity Type Corpus

BBN's NE annotation of the Wall Street Journal corpus (Weischedel and Brunstein, 2005) takes the form of SGML inline markup of text, tokenized to be completely compatible with the Penn Treebank annotation, e.g., *fearlast* and *cannot* are split in the same ways. Named entity categories include: Person, Organization, Location, GPE, Facility, Money, Percent, Time and Date, based on the definitions of these categories in MUC (Chinchor and Robinson, 1998) and ACE[7] tasks. Subcategories are included as well. Note however that from this corpus we only use NE boundaries to derive NAME dependencies between NE tokens, e.g., we create a NAME dependency from *Mary* to *Smith* given the NE mention *Mary Smith*.

3.1.3 Proposition Bank I (PropBank)

The PropBank annotation (Palmer et al., 2005) classifies the arguments of all the main verbs in the Penn Treebank corpus, other than *be*. Arguments are numbered (ARG0, ARG1, . . .) based on lexical entries or frame files. Different sets of arguments are assumed for different rolesets. Dependent constituents that fall into categories independent of the lexical entries are classified as various types of ARGM (TMP, ADV, etc.).[8] Rather than using PropBank directly, we used the version created for the CoNLL-2005 shared task (Carreras and Màrquez, 2005). PropBank's pointers to subtrees are converted into the list of leaves of those subtrees, minus the empty categories. On occasion, arguments of verbs end up being two non-adjacent substrings. For example, the argument of *claims* in the following sentence is indicated in bold: **This sentence**, *Mary claims,* **is self-referential**. The CoNLL-2005 format handles this by marking both strings A1 (*This sentence* and *is self-referential*), but adding a *C-* prefix to the argument tag on the second argument. Another difference between the PropBank annotation and the CoNLL-2005 version of it is their treatments of filler gap constructions involving empty categories. PropBank annotation includes the whole chain of empty categories, as well as the antecedent of the empty category (the filler of the gap). In contrast, the CoNLL-2005 version only includes the filler of the gap and if there is no filler, the argument is omitted, e.g., no ARG0 (subject) for *leave* would be included in *I said to leave* because the subject of leave is unspecified.

3.1.4 NomBank

NomBank annotation (Meyers et al., 2004) uses essentially the same framework as PropBank to annotate arguments of nouns. Differences between PropBank and NomBank stem from differences between noun and verb argument structure; differences in treatment of nouns and verbs in the Penn Treebank; and differences in the sophistication of previous research about noun and verb argument structure. Only the subset of nouns that take arguments are annotated in NomBank and only a subset of the non-argument siblings of nouns are marked as ARGM. These limitations were necessary to make the NomBank task consistent and tractable. In addition, long distance dependencies of nouns, e.g., the relation between *Mary* and *walk* in *Mary took dozens of walks* is handled as follows: *Mary* is marked as the ARG0 of *walk* and *took + dozens + of* is marked as a support chain in NomBank. In contrast, verbal long distance dependencies can be handled by means of empty categories in the Penn Treebank, e.g., the relation be-

[7] `http://projects.ldc.upenn.edu/ace/`

[8] PropBank I is used here. Later versions of PropBank mark instances of *be* in addition to other verbs. PropBank's use of the terms *roleset* and *ARGM* correspond approximately to *sense* and *adjunct* in common usage.

tween *John* and *walked* in *John seemed t to walk*. Support chains are needed because nominal long distance dependencies are not captured under the Penn Treebank's system of empty categories.

3.2 Conversion to Dependencies

3.2.1 Syntactic Dependencies

There exists no large-scale dependency treebank for English, and we thus had to construct a dependency-annotated corpus automatically from the Penn Treebank (Marcus et al., 1993). Since dependency syntax represents grammatical structure by means of labeled binary head–dependent relations rather than phrases, the task of the conversion procedure is to identify and label the head–dependent pairs. The idea underpinning constituent-to-dependency conversion algorithms (Magerman, 1994; Collins, 1999; Yamada and Matsumoto, 2003) is that head–dependent pairs are created from constituents by selecting one word in each phrase as the head and setting all other as its dependents. The dependency labels are then inferred from the phrase–subphrase or phrase–word relations.

Our conversion procedure (Johansson and Nugues, 2007) differs from this basic approach by exploiting the rich structure of the constituent format used in Penn Treebank 3:

- Grammatical function labels that often can be directly used in the dependency framework.
- Long-distance grammatical relations represented by means of empty categories and secondary edges, which can be used to create (often nonprojective) dependency links.

Of the grammatical function tags available in the Treebank, we removed the HLN, NOM, TPC, and TTL tags since they represent structural properties of single phrases rather than binary relations. For compatibility between the WSJ and Brown corpora, we removed the ETC, UNF, and IMP tags from Brown and the CLR tag from WSJ.

Algorithms 1 and 2 show the constituent-to-dependency conversion algorithm and function labeling, respectively. The first steps apply structural transformations to the constituent trees. Next, a head word is assigned to each constituent. After this, grammatical functions are inferred, allowing a dependency tree to be created.

To find head children (used in assign-heads), a system of rules is used (Table 3). The first column in the table indicates the phrase type, the second is the search direction, and the third is a priority list of phrase types to look for. For instance, to find the head of an S phrase, we look from right to left for a VP. If no VP is found, look for anything with a PRD function tag, and so on.

Algorithm 1: Pseudocode for constituent-to-dependency conversion.

```
procedure constituents-to-dependencies($T$)
  import-glarf($T$)
  reattach-traces($T$)
  split-small-clauses($T$)
  assign-heads($T.root$)
  assign-functions($T$)
  return create-dependency-tree($T$)

procedure import-glarf($T$)
  Import a GLARF surface dependency graph $G$
  for each multi-word name $N$ in $G$
    for each token $d$ in $N$
      Set the function tag of $d$ to NAME
  for each dependency link $h \rightarrow_L d$ in $G$
    if $L \in$ { APPOSITE, A-POS, N-POS, POST-HON, Q-POS,
           RED-RELATIVE, SUFFIX, T-POS, TITLE }
    or if $h$ and $d$ are inside a split word
      Set the function tag of $d$ to $L$ in $T$
      if $h$ and $d$ are part of a larger constituent
        Add an NX constituent to $T$ that brackets $h$ and $d$

procedure reattach-traces($T$)
  for each empty category $t$ in $T$
    if $t$ is linked to a constituent $C$ via a secondary edge label $L$
      and $L \in$ { *ICH*, *T*, *RNR* }
      disconnect $C$
      disconnect the secondary edge
      attach $C$ to the parent of $t$

procedure split-small-clauses($T$)
  for each verb phrase $C$ in $T$
    if $C$ has a child $S$ and the phrase label of $S$ is S
    and $S$ is not preceded by a `` or , tag
    and $S$ has a subject child $s$
      disconnect $s$
      attach $s$ to $C$
      set the function tag of $s$ to OBJ
      set the function tag of $S$ to OPRD

procedure assign-heads($N$)
  for each child $C$ of $N$
    assign-heads($C$)
  if is-coordinated($N$)
    $e \leftarrow$ index of first CC or CONJP or , or :
  else
    $e \leftarrow$ index of last child of $N$
  find head child $H$ between 1 and $e$ according to head rules (Table 3)
  $N.head \leftarrow H.head$

procedure is-coordinated($N$)
  if $N$ has the label UCP return True
  if $N$ has a CC or CONJP child which is not leftmost return True
  if $N$ has a , or : child $c$, and $c$ is not leftmost or rightmost or
  crossed by an apposition link, return True
  else return False

procedure create-dependency-tree($T$)
  $D \leftarrow \{\}$
  for each token $t$ in $T$
    let $C$ be the highest constituent that $t$ is the head of
    let $P$ be the parent of $C$
    let $L$ be the function tag of $C$
    $D \leftarrow D \cup P.head \rightarrow_L t$
  return $D$
```

Moreover, since the grammatical structure in-

ADJP	←	NNS QP NN $ ADVP JJ VBN VBG ADJP JJR NP JJS DT FW RBR RBS SBAR RB
ADVP	→	RB RBR RBS FW ADVP TO CD JJR JJ IN NP JJS NN
CONJP	→	CC RB IN
FRAG	→	(NN⋆ | NP) W⋆ SBAR (PP | IN) (ADJP | JJ) ADVP RB
INTJ	←	⋆
LST	→	LS :
NAC, NP, NX, WHNP	←	(NN⋆ | NX) NP-ε JJR CD JJ JJS RB QP NP
PP, WHPP	→	IN TO VBG VBN RP FW
PRN	→	S⋆ N⋆ W⋆ PP|IN ADJP|JJ⋆ ADVP|RB⋆
PRT	→	RP
QP	←	$ IN NNS NN JJ RB DT CD NCD QP JJR JJS
RRC	→	VP NP ADVP ADJP PP
S	←	VP ⋆-PRD S SBAR ADJP UCP NP
SBAR	←	S SQ SINV SBAR FRAG IN DT
SBARQ	←	SQ S SINV SBARQ FRAG
SINV	←	VBZ VBD VBP VB MD VP ⋆-PRD S SINV ADJP NP
SQ	←	VBZ VBD VBP VB MD ⋆-PRD VP SQ
UCP	→	⋆
VP	→	VBD VBN MD VBZ VB VBG VBP VP ⋆-PRD ADJP NN NNS NP
WHADJP	←	CC WRB JJ ADJP
WHADVP	→	CC WRB
X	→	⋆

Table 3: Head rules.

Algorithm 2: Pseudocode for the function labeling procedure.

```
procedure assign-functions($T$)
  for each constituent $C$ in $T$
    if $C$ has no function tag from Penn or GLARF
      $L \leftarrow$ infer-function($C$)
      Set the function tag of $C$ to $L$

procedure infer-function($C$)
  let $c$ be the head of $C$, $P$ the parent of $C$, and $p$ the head of $P$
  if $C$ is an object return OBJ
  if $C$ is PRN return PRN
  if $h$ is punctuation return P
  if $C$ is coordinated with $P$ return COORD
  if $C$ is PP, ADVP, or SBAR and $P$ is VP return ADV
  if $C$ is PRT and $P$ is VP return PRT
  if $C$ is VP and $P$ is VP, SQ, or SINV return VC
  if $C$ is TO and $P$ is VP return IM
  if $P$ is SBAR and $p$ is IN return SUB
  if $P$ is VP, S, SBAR, SBARQ, SINV, or SQ and $C$ is RB return ADV
  if $P$ is NP, NX, NAC, or WHNP return NMOD
  if $P$ is ADJP, ADVP, WHADJP, or WHADVP return AMOD
  if $P$ is PP or WHPP return PMOD
  else return DEP
```

side noun phrases (NP) is under-specified in the Penn Treebank, we imported dependencies inside NPs and hyphenated words from a version of the Penn Treebank mapped into GLARF, the Grammatical and Logical Argument Representation Framework (Meyers et al., 2007).

The parts of GLARF's NP analysis that are most relevant to this task include: (i) identifying apposites (APPO, e.g., that *book* depends on *gift* in *Mary's gift, a book about cheese*; (ii) the identification of name boundaries taken from BBN's NE annotation, e.g., identifying that *Smith* depends on *Mary* which depends on *appointment* in *the Mary Smith appointment*; (iii) identifying TITLE and POSTHON dependencies, e.g., determining that *Ms.* and *III* depend on *Mary* in *Ms. Mary Smith III*. These identifications were carried out by hand-coded rules that have been fine tuned as part of GLARF, over the past several years. For example, identifying apposition constructions requires identifying that both the head and the apposite can stand alone – proper nouns (*John Smith*), plural nouns (*books*), and singular common nouns with determiners (*the book*) are stand-alone cases, whereas singular nouns without determiners (*green book*) do not qualify.

We split Treebank tokens at a hyphen (-) or a forward slash (/) if the segments on either side of these delimiters are: (a) a word in a dictionary (COMLEX Syntax or any of the dictionaries available on the NOMLEX website); (b) part of a markable Named Entity;[9] or (c) a prefix from the list: *co*, *pre*, *post*, *un*, *anti*, *ante*, *ex*, *extra*, *fore*, *non*, *over*, *pro*, *re*, *super*, *sub*, *tri*, *bi*, *uni*, *ultra*. For example, *York-based* was split into 3 segments: (1) *York*, (2) - and (3) *based*.

[9] The CoNLL-2008 website contains a *Named Entity Token gazetteer* to aid in this segmentation.

3.2.2 Semantic Dependencies

When encoding the semantic dependencies, it was necessary to convert the underlying constituent analysis of PropBank and NomBank into a dependency analysis. Because semantic predicates are already assigned to individual tokens in both PropBank (the version used for the CoNLL-2005 shared task) and NomBank, constituent-to-dependency conversion is thus necessary only for semantic arguments. Conceptually, this conversion can be handled using similar heuristics as described in Section 3.2.1. However, in order to avoid replicating this effort and to ensure compatibility between syntactic and semantic dependencies, we decided to generate semantic dependencies using only argument boundaries and the syntactic dependencies generated in Section 3.2.1, i.e., ignoring syntactic constituents. Given this input, we identify the head of a semantic argument using the following heuristic:

> The head of a semantic argument is assigned to the token inside the argument boundaries whose head is a token outside the argument boundaries.

This heuristic works remarkably well: over 99% of the PropBank arguments in the training corpus have a single token whose head is located outside of the argument boundaries. As a simple example, consider the following annotated text: [*sold*]$_{\text{PRED}}$ [*1214 cars*]$_{\text{ARG1}}$ [*in the U.S.*]$_{\text{ARGM-LOC}}$. Using the above heuristic, the head of the ARG1 argument is set to *cars*, because it has an OBJ dependency to *sold*, and the head of the ARGM-LOC argument is set to *in*, because it modifies *sold* through a LOC dependency.

While this heuristic processes the vast majority of arguments, there are several cases that require special treatment. We discuss these situations in the remainder of this section.

Arguments with several syntactic heads

For 0.7% of the semantic arguments, the above heuristic detects several syntactic heads for the given boundary. For example, in the text [*it*]$_{\text{ARG0}}$ [*expects*]$_{\text{PRED}}$ [*its U.S. sales to remain steady at about 1200 cars*]$_{\text{ARG1}}$, the above heuristic assigns two syntactic heads to ARG1: *sales*, which modifies *expects* through an OBJ dependency, and *to*, which modifies *expects* through a PRD dependency. These situations are caused by the constituent-to-dependency conversion process described in Section 3.2.1, which in some cases interprets syntax differently than the original Treebank annotation, e.g., the raising phenomenon for the PRD dependency in the above example. In such cases, we split the original argument into a sequence of discontinuous arguments, e.g., the ARG1 in the above example becomes [*its U.S. sales*]$_{\text{ARG1}}$ [*to remain steady at about 1200 cars*]$_{\text{C-ARG1}}$.

Merging discontinuous arguments

While in the above case we split arguments, there are situations where we can merge arguments that were initially discontinuous in PropBank or NomBank. This typically happens when the PropBank/NomBank predicate is infixed inside one of its arguments. For example, in the text [*Million-dollar conferences*]$_{\text{ARG1}}$ *were* [*held*]$_{\text{PRED}}$ [*to chew on subjects such as...*]$_{\text{C-ARG1}}$, PropBank lists multiple constituents as aggregately filling the ARG1 slot of *held*. These cases are detected automatically because the least common ancestor of the argument pieces is actually one of the argument segments. In the above example, *to chew on subjects such as...* depends on *Million-dollar conferences* because *to* modifies *conferences* through a NMOD dependency. In these situations, we treat the least common ancestor, e.g., *conferences* in the above text, as the true argument. This heuristic allowed us to merge 1665 (or 0.6% of total) arguments that were initially discontinuous in the PropBank training corpus.

Empty categories

PropBank and NomBank both encode chains of empty categories. As with the 2005 shared task (Carreras and Màrquez, 2005), we used the head of the antecedent of empty categories as arguments rather than empty categories. Furthermore, empty category arguments with no antecedents were ignored.[10] For example, given *The man wanted t to make a speech*, we assume that the A0 of *make* and *speech* is *man*, rather than the chain consisting of the empty category represented as *t* and *man*.

Annotation disagreements

NomBank and Penn Treebank annotators sometimes disagree about constituent structure. Nom-

[10] Under our approach to filler gap constructions, the filler is a shared argument (as in Relational Grammar, most Feature Structure and Dependency Grammar frameworks), in contrast with the Penn Treebank's empty category antecedent approach (more closely resembling the various Chomskian approaches).

Label	Freq.	Description
NMOD	324834	Modifier of nominal
P	135260	Punctuation
PMOD	115988	Modifier of preposition
SBJ	89371	Subject
OBJ	66677	Object
ROOT	49178	Root
ADV	47379	General adverbial
NAME	41138	Name-internal link
VC	35250	Verb chain
COORD	31140	Coordination
DEP	29456	Unclassified
TMP	26305	Temporal adverbial or nominal modifier
CONJ	24522	Second conjunct (dependent on conjunction)
LOC	18500	Locative adverbial or nominal modifier
AMOD	17868	Modifier of adjective or adverbial
PRD	16265	Predicative complement
APPO	16163	Apposition
IM	16071	Infinitive verb (dependent on infinitive marker *to*)
HYPH	14073	Token part of a hyphenated word (dependent on a preceding part of the hyphenated word)
HMOD	13885	Token inside a hyphenated word (dependent on the head of the hyphenated word)
SUB	12995	Subordinated clause (dependent on subordinating conjuction)
OPRD	11707	Predicative complement of raising/control verb
SUFFIX	10548	Possessive suffix (dependent on possessor)
DIR	6145	Adverbial of direction
TITLE	5917	Title (dependent on name)
MNR	4753	Adverbial of manner
POSTHON	4377	Posthonorific modifier of nominal
PRP	4013	Adverbial of purpose or reason
PRT	3235	Particle (dependent on verb)
LGS	3115	Logical subject of a passive verb
EXT	2374	Adverbial of extent
PRN	2176	Parenthetical
EXTR	658	Extraposed element in cleft
DTV	496	Dative complement (*to*) in dative shift
PUT	271	Complement of the verb *put*
BNF	44	Benefactor complement (*for*) in dative shift
VOC	24	Vocative

Table 4: Statistics for atomic syntactic labels.

Bank annotators are in effect assuming that the constituents provided form a phrase. In this case, the constituents are adjacent to each other. For example, consider the NP *the human rights discussion*. In this case, the Penn Treebank would treat each of the four words *the*, *human*, *rights*, *discussion* as daughters of a single NP node. However, NomBank would treat *human rights* as a single ARG1 of *discussion*. Since noun noun modification constructions are head final, we can easily determine (via GLARF) that *rights* is the markable dependent of *discussion*.

Support chains

Finally, NomBank's encoding of support chains is handled as chains of dependencies in the data (although these are not scored). For example, given *Mary took dozens of walks*, where *Mary* is the ARG0 of *walks*, the support chain *took* + *dozens* + *of* is represented as a sequence of dependencies: *of* depends on *Mary*, *dozens* depends on *of* and *took* depends on *dozens*. Each of these dependencies is labeled *SU*.

3.3 Overview of Corpora

The syntactic dependency types are divided into *atomic* types that consist of a single label, and *non-atomic* types consisting of more than one label. There are 38 atomic and 70 non-atomic labels in the corpus. There are three types of non-atomic labels: those consisting of a PRD or OPRD concatenated with an adverbial label such as LOC or TMP; gapping labels such as GAP-SBJ; and combined adverbial tags such as LOC-TMP.

Table 4 shows statistics for the atomic syntactic dependencies: label type, the frequency of the label in the complete corpus, and a description of the label. Table 5 shows the corresponding statistics for non-atomic dependencies, excluding gapping dependencies. The non-atomic labels are rare, which made it difficult to learn these relations ef-

Label	Frequency
LOC-PRD	798
PRD-TMP	51
PRD-PRP	45
LOC-OPRD	31
DIR-PRD	4
MNR-PRD	3
LOC-TMP	2
MNR-TMP	1
LOC-MNR	1
DIR-OPRD	1

Table 5: Statistics for non-atomic syntactic labels excluding gapping labels.

Label	Frequency
GAP-SBJ	116
GAP-OBJ	102
DEP-GAP	83
GAP-TMP	69
GAP-PRD	66
GAP-LGS	44
GAP-LOC	42
DIR-GAP	37
GAP-PMOD	22
GAP-VC	20
EXT-GAP	16
ADV-GAP	15
GAP-NMOD	13
GAP-LOC-PRD	6
DTV-GAP	6
AMOD-GAP	6
GAP-MNR	5
GAP-PRP	4
EXTR-GAP	3
GAP-SUB	1
GAP-PUT	1
GAP-OPRD	1

Table 6: Statistics for non-atomic labels containing a gapping label.

fectively. Table 6 shows the table for non-atomic labels containing a gapping label.

A dependency link $w_i \rightarrow w_j$ is said to be *projective* if all words occurring between w_i and w_j in the surface word order are dominated by w_i (where dominance is the transitive closure of the direct link relation). Nonprojective links are impossible to handle for the search procedures in many types of dependency parsers. It has been previously observed that the majority of dependencies in all languages are projective, and this is particularly true for English – in the complete corpus, only 4118 links (0.4%) are nonprojective. 3312 sentences, or 7.6%, contain at least one nonprojective link.

Table 7 shows statistics for different types of nonprojective links: nonprojectivity caused by *wh-movement*, such as in *Where are you going?* or *What have you done?*; split clauses such as *Even to make love, he says, you need experience*; split noun phrases such as *hold a hearing tomorrow on the topic*; and all other types of nonprojective links.

Type	Frequency
wh-movement	1709
Split clause	734
Split noun phrase	590
Other	1085

Table 7: Statistics for nonprojective links.

POS	Frequency
NN	68477
NNS	30048
VBD	24106
VB	23650
VBN	19339
VBG	14245
VBZ	10883
VBP	6330
Other	83

Table 8: Statistics for predicates, by POS tags.

Lastly, Tables 8 and 9 summarizes statistics for semantic predicates and roles. Table 8 shows the number of non-support predicates with a given POS tag in the whole corpus (we used GPOS or PPOSS for predicates inside hyphenated words). The last line shows the number of predicates with a POS tag that does not start with NN or VB. This last table entry is generated by POS tagger mistakes when producing the PPOSS tags, or by errors in our NomBank/PropBank conversion software.[11] Nevertheless, the overall picture given by the table indicates that predicates are almost perfectly distributed between nouns and verbs: there are 98525 nominal and 98553 verbal predicates.

Table 9 shows the number of arguments with a given role label. For brevity we list only labels that are instantiated at least 10 times in the whole corpus. The total number of arguments labeled with a role label with frequency lower than 10 is listed in the last line in the table. The table indicates that, while the top three most common role labels are "core" labels (`A1`, `A0`, `A2`), modifier arguments (`AM-*`) account for approximately 20% of the total number of arguments. On the other hand, discontinuous arguments are not common: only 0.7% of the total number of arguments have a continuation label (`C-*`).

[11] In very few situations, we select incorrect head tokens for multi-word predicates.

Label	Frequency
A1	161409
A0	109437
A2	51197
AM-TMP	25913
AM-MNR	13080
AM-LOC	11409
A3	10269
AM-MOD	9986
AM-ADV	9496
AM-DIS	5369
R-A0	4432
AM-NEG	4097
A4	3281
C-A1	3118
R-A1	2565
AM-PNC	2445
AM-EXT	1428
AM-CAU	1346
AM-DIR	1318
R-AM-TMP	797
R-A2	307
R-AM-LOC	246
R-AM-MNR	155
A5	91
AM-PRD	78
C-A0	70
C-A2	65
R-AM-CAU	50
C-A3	37
R-A3	29
C-AM-MNR	24
C-AM-ADV	20
AM-REC	16
AA	14
R-AM-PNC	12
C-AM-EXT	11
C-AM-TMP	11
C-A4	11
Frequency $<$ 10	70

Table 9: Statistics for semantic roles.

4 Submissions and Results

Nineteen groups submitted test runs in the closed challenge and five groups participated in the open challenge. Three of the latter groups participated only in the open challenge, and two of these submitted results only for the semantic subtask. These results are summarized in Tables 10 and 11.

Table 10 summarizes the official results – i.e., results at evaluation deadline – for the closed challenge. Note that several teams corrected bugs and/or improved their systems and they submitted post-evaluation scores (accounted in the shared task website). The table indicates that most of the top results cluster together: three systems had a labeled macro F_1 score on the WSJ+Brown corpus around 82 points (che, ciaramita, and zhao); five systems scored around 79 labeled macro F_1 points (yuret, samuelsson, zhang, henderson, and watanabe). Remarkably, the top-scoring system (johansson) is in a class of its own, with scores 2–3 points higher than the next system. This is most likely caused by the fact that Johansson and Nugues (2008) implemented a thorough system that addressed all facets of the task with state-of-the-art methods: second-order parsing model, argument identification/classification models separately tuned for PropBank and NomBank, reranking inference for the SRL task, and, finally, joint optimization of the complete task using meta-learning (more details in Section 5).

Table 11 lists the official results in the open challenge. The results in this challenge are lower than in the closed challenge, but this was somewhat to be expected considering that there were fewer participants in this challenge and none of the top five groups in the closed challenge submitted results in the open challenge. Only one of the systems that participated in both challenges (zhang) improved the results submitted in the closed challenge. Zhang et al. (2008) achieved this by extracting features for their semantic subtask models both from the parser used in the closed challenge and a secondary parser that was trained on a different corpus. The improvements measured were relatively small for the in-domain WSJ corpus (0.2 labeled macro F_1 points) but larger for the out-of-domain Brown corpus (approximately 1 labeled macro F_1 point).

Tables 10 and 11 indicate that in both challenges the results on the out-of-domain corpus (Brown) are much lower than the results measured in-domain (WSJ). The difference is around 7–8 LAS points for the syntactic subtask and 12–14 labeled F_1 points for semantic dependencies. Overall, this yields a drop of approximately 10 labeled macro F_1 points for most systems. This performance decrease on out-of-domain corpora is consistent with the results reported in CoNLL-2005 on SRL (using the same Brown corpus). These results indicate that domain adaptation is a problem that is far from being solved for both syntactic and semantic analysis of text. Furthermore, as the scores on the syntactic and semantic subtasks indicate, domain adaptation becomes even harder as the task to be solved gets more complex.

We describe the participating systems in the next section. Then, in Section 6, we revert to result analysis using different evaluation measures and different views of the data.

	Labeled Macro F_1 (complete task)			Labeled Attachment Score (syntactic dependencies)			Labeled F_1 (semantic dependencies)		
	WSJ+Brown	WSJ	Brown	WSJ+Brown	WSJ	Brown	WSJ+Brown	WSJ	Brown
johansson	**84.86** (1)	**85.95**	**75.95**	**89.32** (1)	**90.13**	**82.81**	**80.37** (1)	**81.75**	**69.06**
che	82.66 (2)	83.78	73.57	86.75 (5)	87.51	80.73	78.52 (2)	80.00	66.37
ciaramita	82.06 (3)	83.25	72.46	86.60 (11)	87.47	79.67	77.50 (3)	79.00	65.24
zhao	81.44 (4)	82.62	71.78	86.66 (8)	87.52	79.83	76.16 (4)	77.67	63.69
yuret	79.84 (5)	80.97	70.55	86.62 (10)	87.39	80.46	73.06 (5)	74.54	60.62
samuelsson	79.79 (6)	80.92	70.49	86.63 (9)	87.36	80.77	72.94 (6)	74.47	60.18
zhang	79.32 (7)	80.41	70.48	87.32 (2)	88.14	80.80	71.31 (7)	72.67	60.16
henderson	79.11 (8)	80.19	70.34	86.91 (4)	87.78	80.01	70.97 (8)	72.26	60.38
watanabe	79.10 (9)	80.30	69.29	87.18 (3)	88.06	80.17	70.84 (9)	72.37	58.21
morante	78.43 (10)	79.52	69.55	86.07 (12)	86.88	79.58	70.51 (10)	71.88	59.23
li	78.35 (11)	79.38	70.01	86.69 (6)	87.42	80.8	69.95 (11)	71.27	59.17
baldridge	77.49 (12)	78.57	68.53	86.67 (7)	87.42	80.64	67.92 (14)	69.35	55.95
chen	77.00 (13)	77.95	69.23	84.47 (16)	85.20	78.58	69.45 (12)	70.62	59.81
lee	76.90 (14)	77.96	68.34	84.82 (15)	85.69	77.83	68.71 (13)	69.95	58.63
sun	76.28 (15)	77.10	69.58	85.75 (13)	86.37	80.75	66.61 (15)	67.62	58.26
choi	71.23 (16)	72.22	63.44	77.56 (17)	78.58	69.46	64.78 (16)	65.72	57.4
trandabat	63.45 (17)	64.21	57.41	85.21 (14)	85.96	79.24	40.63 (17)	41.36	34.75
lluis	63.29 (18)	63.74	59.65	71.95 (18)	72.30	69.14	54.52 (18)	55.09	49.95
neumann	19.93 (19)	20.13	18.14	16.25 (19)	16.22	16.47	22.36 (19)	22.86	17.94

Table 10: Official results in the closed challenge (post-evaluation scores are available on the shared task website). Teams are denoted by the last name of the first author of the corresponding paper in the proceedings or the last name of the person who registered the team if no paper was submitted. Italics indicate that there is no corresponding paper in the proceedings. Results are sorted in descending order of the labeled macro F_1 score on the WSJ+Brown corpus. The number in parentheses next to the WSJ+Brown scores indicates the system rank in the corresponding task.

	Labeled Macro F_1 (complete task)			Labeled Attachment Score (syntactic dependencies)			Labeled F_1 (semantic dependencies)		
	WSJ+Brown	WSJ	Brown	WSJ+Brown	WSJ	Brown	WSJ+Brown	WSJ	Brown
vickrey	–	–	–	–	–	–	**76.17** (1)	**77.38**	**66.23**
riedel	–	–	–	–	–	–	74.59 (2)	75.72	65.38
zhang	**79.61** (1)	**80.61**	**71.45**	**87.32** (1)	**88.14**	**80.80**	71.89 (3)	73.08	62.11
li	77.84 (2)	78.87	69.51	86.69 (2)	87.42	**80.80**	68.99 (4)	70.32	58.22
wang	76.19 (3)	78.39	59.89	84.56 (3)	85.50	77.06	67.12 (5)	70.41	42.67

Table 11: Official results in the open challenge (post-evaluation scores are available on the shared task website). Teams are denoted by the last name of the first author of the corresponding paper in the proceedings or the last name of the person who registered the team if no paper was submitted. Italics indicate that there is no corresponding paper in the proceedings. Results are sorted in descending order of the labeled F_1 score for semantic dependencies on the WSJ+Brown corpus. The number in parentheses next to the WSJ+Brown scores indicates the system rank in the corresponding task.

5 Approaches

Table 5 summarizes the properties of the systems that participated in the closed the open challenges. The second column of the table highlights the overall architectures. We used + to indicate that the components are sequentially connected. The lack of a + sign indicates that the corresponding tasks are performed jointly. For example, Riedel and Meza-Ruiz (2008) perform predicate and argument identification and classification jointly, whereas Ciaramita et al. (2008) implemented a pipeline architecture of three components. We use the || to indicate that several different architectures that span multiple subtasks were deployed in parallel.

This summary of system architectures indicates that it is common that systems combine several components in the semantic or syntactic subtasks – e.g., nine systems jointly performed predicate/argument identification and classification – but only four systems combined components between the syntactic and semantic subtasks: Henderson et al. (2008), who implemented a generative history-based model (Incremental Sigmoid Belief Networks with vectors of latent variables) where syntactic and semantic structures are separately generated but using a synchronized derivation (sequence of actions); Samuelsson et al. (2008), who, within an ensemble-based architecture, implemented a joint syntactic-semantic model using MaltParser with labels enriched with semantic information; Lluís and Màrquez, who used a modified version of the Eisner algorithm to jointly predict syntactic and semantic dependencies; and finally, Sun et al. (2008), who integrated dependency label classification and argument identification using a maximum-entropy Markov model. Additionally, Johansson and Nugues (2008), who had the highest ranked system in the closed challenge, integrate syntactic and semantic analysis in a final reranking step, which maximizes the joint syntactic-semantic score in the top k solutions. In the same spirit, Chen et al. (2008) search in the top k solutions for the one that maximizes a global measure, in this case the joint probability of the complete problem. These joint learning strategies are summarized in the **Joint Learning/Opt.** column in the table. The system of Riedel and Meza-Ruiz (2008) deserves a special mention: even though Riedel and Meza-Ruiz did not implement a syntactic parser, they are the only group that performed the complete SRL subtask – i.e., predicate identification and classification, argument identification and classification – jointly, simultaneously for all the predicates in a sentence. They implemented a joint SRL model using Markov Logic Networks and they selected the overall best solution using inference based on the cutting-plane algorithm.

Although some of the systems that implemented joint approaches obtained good results, the top five systems in the closed challenge are essentially systems with pipeline architectures. Furthermore, Johansson and Nugues (2008) and Riedel and Meza-Ruiz (2008) showed that joint learning/optimization improves the overall results, but the improvement is not large. These initial efforts indicate at least that the joint modeling of this problem is not a trivial task.

The **D Arch.** and **D Inference** columns summarize the parsing architectures and the corresponding inference strategies. Similar to last year's shared task (Nivre et al., 2007), the vast majority of parsing models fall in two classes: transition-based ("trans" in the table) or graph-based ("graph") models. By and large, transition-based models use a greedy inference strategy, whereas graph-based models used different Maximum Spanning Tree (MST) algorithms: Carreras (2007) – MST^C, Eisner (2000) – MST^E, or Chu-Liu/Edmonds (McDonald et al., 2005; Chu and Liu, 1965; Edmonds, 1967) – MST$^{CL/E}$. More interestingly, most of the best systems used some strategy to mitigate parsing errors. In the top three systems in the closed challenge, two (che and ciaramita) used parser combination through voting and/or stacking of different models (see the **D Comb.** column). Samuelsson et al. (2008) perform a MST inference with the bag of all dependencies output by the individual MALT parser variants. Johansson and Nugues (2008) use a single parsing model, but this model is extended with second-order features.

The **PA Arch.** and **PA Inference** columns summarize the architectures and inference strategies used for the identification and classification of predicates and arguments. The columns indicate that most systems modeled the SRL problem as a token-by-token classification problem ("class" in the table) with a corresponding greedy inference strategy. Some systems (e.g., yuret, samuelsson, henderson, lluis) incorporate SRL within parsing, in which case we report the corresponding parsing architecture and inference approach. Vickrey and Koller (2008) simplify the sentences to be labeled using a set of hand-crafted rules before deploying a classification model on top of a constituent-based representation. Unlike in the case of parsing, few systems (yuret, samuelssson, and morante) combine several PA models and the combination is limited to simple voting strategies (see the **PA Comb.** column).

Finally, the **ML Methods** column lists the Machine Learning (ML) methods used. The column indicates that maximum entropy (ME) was the most popular method (12 distinct systems relied on it). Support Vector Machines (SVM) (eight systems) and the Perceptron algorithm (three systems) were also popular ML methods.

6 Analysis

Section 4 summarized the results in the closed and open challenges using the official evaluation measures. In this section, we analyze the submitted runs using different evaluation measures, e.g., Exact Match or Perfect Proposition F_1 scores, and different views of the data, e.g., only non-projective dependencies or NomBank versus PropBank frames.

closed	**Overall Arch.**	**D Arch.**	**D Comb.**	**D Inference**	**PA Arch.**	**PA Comb.**	**PA Inference**	**Joint Learning/Opt.**	**ML Methods**
johansson	D+PI+PC+AI+AC	graph	no	MST^{C}	class	no	rerank	rerank	Perceptron, ME
che	D+PI+PC+AIC	graph	stacking	MST$^{CL/E}$	class	no	ILP	no	ME
ciaramita	D+PIC+AIC	trans	voting, stacking	greedy	class	no	rerank	no	SVM, ME, Perceptron
zhao	D+AIC+PI+PC	trans	no	greedy	class	no	greedy	no	ME
yuret	D+(PIC+AI+AC $\Vert$ PIC+AIC)	graph	no	MST^{E}	class, generative	voting	greedy	no	MLE, MBL
samuelsson	D+PI+ (AI+AC $\Vert$ DAIC)+PC	trans	MST$^{CL/E}$ blending	greedy	class, trans	voting	greedy	unified labels	SVM
zhang	D+PI+AI+AC+PC	graph, trans	meta-learning	MST$^{CL/E}$, greedy	class	no	greedy	no	SVM,ME
henderson	DPAIC+D	generative, trans	no	beam search	trans	no	beam search	synchronized derivation	ISBN
watanabe	DI+DC+PI+PC+AI+AC	relative preference model	no	greedy tournament model, Viterbi	class	no	no	no	SVM, CRF, MBL
morante	D+PI+AI+AC	trans	no	greedy	class	voting	greedy	no	SVM, MBL
li	D+PIC+AIC	graph	no	MST$^{CL/E}$	class	voting	greedy	no	ME
chen	D+PI+PC+AIC	trans	no	prob	class	no	prob	global probability optimization	ME
lee	D+PI+AIC+PC	trans	no	greedy	class	no	prob	no	SVM, ME
sun	DI+PI+DCAI+AC	graph	no	MST^{E}, Viterbi	graph	no	Viterbi, ILP	MEMM, Viterbi	ME
lluis	D+PI+DAIC+PC	graph	no	MST^{E}	graph	no	MST^{E}	MST^{E}	Perceptron, SVM
neumann	D+PI+PC+AI+AC	trans	no	greedy	class	no	no	no	ME
open									
vickrey	AI+AC+PI+PC	–	–	–	sentence simplification, class	no	greedy	–	ME
riedel	PAIC	–	–	–	Markov Logic Network	no	Cutting Plane	–	MIRA
wang	PI+AIC	trans, graph	no	greedy, MST$^{CL/E}$	class	no	prob	no	SVM, ME MIRA

Table 12: Summary of system architectures that participated in the closed and open challenges. The closed-challenge systems are sorted by macro labeled F_1 score on the WSJ+Brown corpus. Because some open-challenge systems did not implement syntactic parsing, these systems are sorted by labeled F_1 score of the semantic dependencies on the WSJ+Brown corpus. Only the systems that have a corresponding paper in the proceedings are included. Systems that participated in both challenges are listed only in the closed challenge. Acronyms used: **D** - syntactic dependencies, **P** - predicate, **A** - argument, I - identification, **C** - classification. **Overall arch.** stands for the complete system architecture; **D Arch.** stands for the architecture of the syntactic parser; **D Comb.** indicates if the final parser output was generated using parser combination; **D Inference** stands for the type of inference used for syntactic parsing; **PA Arch.** stands the type of architecture used for PAIC; **PA Comb.** indicates if the PA output was generated through system combination; **PA Inference** stands for the the type of inference used for PAIC; **Joint Learning/Opt.** indicates if some form of joint learning or optimization was implemented for the syntactic + semantic global task; **ML methods** lists the ML methods used throughout the complete system.

	Exact Match (complete task)			Perfect Proposition F$_1$ (semantic dependencies)		
closed	WSJ+Brown	WSJ	Brown	WSJ+Brown	WSJ	Brown
johansson	**12.46** (1)	**12.46**	**12.68**	**54.12** (1)	**56.12**	**36.90**
che	10.37 (2)	10.21	11.50	48.05 (2)	50.15	30.90
ciaramita	9.27 (3)	9.04	10.80	46.05 (3)	48.05	28.61
zhao	9.20 (4)	9.00	10.56	43.19 (4)	45.23	26.14
henderson	8.11 (5)	7.75	10.33	39.24 (5)	40.64	27.51
watanabe	7.79 (6)	7.54	9.39	36.44 (6)	38.09	22.72
yuret	7.65 (7)	7.33	9.62	34.61 (9)	36.13	21.78
zhang	7.40 (8)	7.46	7.28	34.96 (8)	36.25	24.22
li	7.12 (9)	6.71	9.62	32.08 (10)	33.45	20.62
samuelsson	6.94 (10)	6.62	8.92	35.20 (7)	36.96	20.22
chen	6.83 (11)	6.46	9.15	31.02 (12)	32.08	22.14
lee	6.69 (12)	6.29	9.15	31.40 (11)	32.52	22.18
morante	6.44 (13)	6.04	8.92	30.41 (14)	31.97	17.49
sun	5.38 (14)	4.96	7.98	30.43 (13)	31.51	21.40
baldridge	5.24 (15)	4.92	7.28	25.35 (15)	26.57	15.26
choi	3.33 (16)	3.50	2.58	24.77 (16)	25.71	17.37
trandabat	3.26 (17)	3.08	4.46	6.59 (18)	6.81	4.76
lluis	2.55 (18)	1.96	6.10	16.07 (17)	16.46	13.00
neumann	0.11 (19)	0.12	0.23	0.30 (19)	0.31	0.20
open						
vickrey	–	–	–	**44.94** (1)	**46.68**	30.28
riedel	–	–	–	42.77 (2)	44.18	**31.15**
zhang	**8.14** (1)	**8.04**	8.92	35.46 (3)	36.74	24.84
li	6.90 (2)	6.46	**9.62**	29.91 (4)	31.30	18.41
wang	5.17 (3)	5.12	5.63	18.63 (5)	20.31	7.09

Table 13: Exact Match and Perfect Proposition F$_1$ scores for runs submitted in the closed and open challenges. The closed-challenge systems are sorted in descending order of Exact Match scores on the WSJ+Brown corpus. Open-challenge submissions are sorted in descending order of the Perfect Proposition F$_1$ score. The number in parentheses next to the WSJ+Brown scores indicates the system rank according to the corresponding scoring measure.

6.1 Exact Match and Perfect Propositions

Table 13 lists the Exact Match and Perfect Proposition F$_1$ scores for test runs submitted in both challenges. Both these scores measure the capacity of a system to correctly parse structures with granularity much larger than a simple dependency, i.e., entire sentences for Exact Match and complete propositions for Perfect Proposition F$_1$ (see Section 2.2.2 for a formal definition of these evaluation measures). The table indicates that these values are much smaller than the scores previously reported, e.g., labeled macro F$_1$. This is to be expected: the probability of an incorrectly parsed unit (sentence or proposition) is much larger given its granularity. However, the main purpose of this analysis is to investigate if systems that focused on joint learning or optimization performed better than others with respect to these global measures. This indeed seems to be the case for at least two systems. The system of Johansson and Nugues (2008), which jointly optimizes the labeled F$_1$ score (for semantic dependencies) and then the labeled macro F$_1$ score (for the complete task), increases its distance from the next ranked system: its Perfect Proposition F$_1$ score is over 6 points higher than the score of the second system in Table 13. The system of Henderson et al. (2008), which was designed for joint learning of the complete task, improves its rank from eighth to fifth compared to the official results (Table 10).

6.2 Nonprojectivity

Table 14 shows the unlabeled F1 scores for prediction of nonprojective syntactic dependencies. Since nonprojectivity is quite rare, many teams chose to ignore this issue. The table shows only those systems that submitted well-formed dependency trees, and whose output contained at least one nonprojective link. The small number of nonprojective links in the training set makes it hard to learn to predict such links, and this is also reflected in the figures. In general, the figures for nonprojective *wh*-movements and split clauses are higher, and they are also the most common types. Also, they are detectable by fairly simple patterns, such as the presence of a *wh*-word or a pair of commas.

System	All	*wh*-mov.	SpCl	SpNP
choi	25.43	49.49	45.47	8.72
lee	46.26	50.30	64.84	20.69
nugues	46.15	58.96	59.26	11.32
samuelsson	24.47	38.15	0	9.83
titov	42.32	50.56	48.71	0
zhang	13.39	5.71	12.33	7.3

Table 14: Unlabeled F1-measures for nonprojective links. Results are given for all links, *wh*-movements, split clauses, and split noun phrases.

6.3 Normalized SRL Performance

Table 6.3 lists the scores for the semantic subtask measured as the ratio of the labeled F_1 score and LAS. As previously mentioned, this score estimates the performance of the SRL component independent of the performance of the syntactic parser. This analysis is not a substitute for the actual experiment where the SRL components are evaluated using correct syntactic information but, nevertheless, it indicates several interesting facts. First, the ranking of the top three systems in Table 10 changes: the system of Che et al. (2008) is now ranked first, and the system of Johansson and Nugues (2008) is second. This shows that Che et al. have a relatively stronger SRL component, whereas Johansson and Nugues developed a better parser. Second, several other systems improved their ranking compared to Table 10: e.g., chen from position thirteenth to ninth and choi from sixteenth to eighth. This indicates that these systems were penalized in the official ranking mainly due to the relative poor performance of their parsers.

Note that this experiment is relevant only for systems that implemented pipeline architectures, where the semantic components are in fact separated from the syntactic ones; this excludes the systems that blended syntax with SRL: henderson, sun, and lluis. Furthermore, systems that had significantly lower scores in syntax will receive an unreasonable boost in ranking according to this measure. Fortunately, there was only one such outlier in this evaluation (neumann), shown in gray in the table.

6.4 PropBank versus NomBank

Table 16 lists the labeled F_1 scores for semantic dependencies for two different views of the testing data sets: for propositions centered around verbal predicates, i.e., from PropBank, and for propositions centered around nominal predicates, i.e., from NomBank.

	Labeled F_1 / LAS		
closed	WSJ+Brown	WSJ	Brown
neumann	137.60 (1)	140.94	108.93
che	90.51 (2)	91.42	82.21
johansson	89.98 (3)	90.70	83.40
ciaramita	89.49 (4)	90.32	81.89
zhao	87.88 (5)	88.75	79.78
yuret	84.35 (6)	85.30	75.34
samuelsson	84.20 (7)	85.24	74.51
choi	83.52 (8)	83.63	82.64
chen	82.22 (9)	82.89	76.11
morante	81.92 (10)	82.73	74.43
zhang	81.67 (11)	82.45	74.46
henderson	81.66 (12)	82.32	75.47
watanabe	81.26 (13)	82.18	72.61
lee	81.01 (14)	81.63	75.33
li	80.69 (15)	81.53	73.23
baldridge	78.37 (16)	79.33	69.38
sun	77.68 (17)	78.29	72.15
lluis	75.77 (18)	76.20	72.24
trandabat	47.68 (19)	48.12	43.85
open			
zhang	82.33	82.91	76.87
li	79.58	80.44	72.05
wang	79.38	82.35	55.37

Table 15: Ratio of the labeled F_1 score for semantic dependencies and LAS for syntactic dependencies. Systems are sorted in descending order of this ratio score on the WSJ+Brown corpus. We only show systems that participated in both the syntactic and semantic subtasks.

The table indicates that, generally, systems performed much worse on nominal predicates than on verbal predicates. This is to be expected considering that there is significant body of previous work that analyzes the SRL problem on PropBank, but minimal work for NomBank. On average, the difference between the labeled F_1 scores for verbal predicates and nominal predicates on the WSJ+Brown corpus is 7.84 points. Furthermore, the average difference between labeled F_1 scores on the Brown corpus alone is 12.36 points. This indicates that the problem of SRL for nominal predicates is more sensitive to domain changes than the equivalent problem for verbal predicates. Our conjecture is that, because there is very little syntactic structure between nominal predicates and their arguments, SRL models for nominal predicates select mainly lexical features, which are more brittle than syntactic or other non-lexicalized features.

Remarkably, there is one system (baldridge) which performed better on the WSJ+Brown for nominal predicates than verbal predicates. Unfortunately, this group did not submit a system-description paper so it is not clear what was their approach.

	Labeled F_1 (verbal predicates)			Labeled F_1 (nominal predicates)		
closed	WSJ+Brown	WSJ	Brown	WSJ+Brown	WSJ	Brown
johansson	**84.45** (1)	**86.37**	**71.87**	74.32 (2)	75.42	**60.13**
che	80.46 (2)	82.17	69.33	**75.18** (1)	**76.64**	56.87
ciaramita	80.15 (3)	82.09	67.62	73.17 (4)	74.42	57.69
zhao	77.67 (4)	79.40	66.38	73.28 (3)	74.69	54.81
samuelsson	76.17 (5)	78.03	64.00	68.13 (7)	69.58	49.24
yuret	75.91 (6)	77.88	63.02	68.81 (5)	69.98	53.58
zhang	74.82 (7)	76.62	63.15	65.61 (11)	66.82	50.18
li	74.36 (8)	76.14	62.92	62.61 (14)	63.76	47.09
henderson	73.80 (9)	75.40	63.36	66.26 (10)	67.44	50.73
watanabe	73.06 (10)	75.02	60.34	67.15 (8)	68.37	50.92
sun	72.97 (11)	74.45	63.50	58.68 (15)	59.73	45.75
morante	72.81 (12)	74.36	62.72	66.50 (9)	67.92	47.97
lee	72.34 (13)	74.15	60.49	62.83 (13)	63.66	52.18
chen	72.02 (14)	73.49	62.46	65.02 (12)	66.14	50.48
choi	70.00 (15)	71.28	61.71	56.16 (16)	57.19	44.05
baldridge	67.02 (16)	68.64	56.50	68.57 (6)	69.78	52.96
lluis	62.42 (17)	63.49	55.49	42.15 (17)	42.81	34.22
trandabat	42.88 (18)	43.79	37.06	37.14 (18)	37.89	27.50
neumann	22.87 (19)	23.53	18.24	21.7 (19)	22.04	17.14
open						
vickrey	**78.41** (1)	**79.75**	**69.57**	**71.86** (1)	**73.29**	53.25
riedel	77.13 (2)	78.72	66.75	70.25 (2)	71.03	**60.17**
zhang	75.00 (3)	76.62	64.44	66.76 (3)	67.79	53.76
li	73.74 (4)	75.57	62.05	61.24 (5)	62.38	46.36
wang	67.50 (5)	70.34	49.72	66.53 (4)	69.83	28.96

Table 16: Labeled F_1 scores for frames centered around verbal and nominal predicates. The number in parentheses next to the WSJ+Brown scores indicates the system rank in the corresponding data set.

Systems can mitigate the inherent differences between verbal and nominal predicates with different models for the two sub-problems. This was indeed the approach taken by two out of the top three systems (johansson and che). Johansson and Nugues (2008) developed different models for verbal and nominal predicates and implemented separate feature selection processes for each model. Che et al. (2008) followed the same method but they also implemented separate domain constraints for inference for the two models.

7 Conclusion

The previous four CoNLL shared tasks popularized and, without a doubt, boosted research in semantic role labeling and dependency parsing. This year's shared task introduces a new task that essentially unifies the problems addressed in the past four years under a unique, dependency-based formalism. This novel task is attractive both from a research perspective and an application-oriented perspective:

- We believe that the proposed dependency-based representation is a better fit for many applications (e.g., Information Retrieval, Information Extraction) where it is often sufficient to identify the dependency between the predicate and the head of the argument constituent rather than extracting the complete argument constituent.

- It was shown that the extraction of syntactic and semantic dependencies can be performed with state-of-the-art performance in linear time (Ciaramita et al., 2008). This can give a significant boost to the adoption of this technology in real-world applications.

- We hope that this shared task will motivate several important research directions. For example, is the dependency-based representation better for SRL than the constituent-based formalism? Does joint learning improve syntactic and semantic analysis?

- Surface (string related patterns, syntax, etc.) linguistic features can often be detected with greater reliability than deep (semantic) features. In contrast, deep features can cover more ground because they regularize across differences in surface strings. Machine learning systems can be more effective by using evidence from both deep and surface features jointly (Zhao, 2005).

Even though this shared task was more complex than the previous shared tasks, 22 different teams submitted results in at least one of the challenges. Building on this success, we hope to expand this effort in the future with evaluations on multiple languages and on larger out-of-domain corpora.

Acknowledgments

We want to thank the following people who helped us with the generation of the data sets: Jesús Giménez, for generating the predicted POS tags with his SVMTool POS tagger, and Massimiliano Ciaramita, for generating columns 1, 2 and 3 in the open-challenge corpus with his semantic tagger.

We also thank the following people who helped us with the organization of the shared task: Paola Merlo and James Henderson for the idea and the implementation of the Exact Match measure, Sebastian Riedel for his dependency visualization software,[12] Hai Zhao, for the the idea of the F_1 ratio score, and Carlos Castillo, for help with the shared task website. Last but not least, we thank the organizers of the previous four shared tasks: Sabine Buchholz, Xavier Carreras, Ryan McDonald, Amit Dubey, Johan Hall, Yuval Krymolowski, Sandra Kübler, Erwin Marsi, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. This shared task would not have been possible without their previous effort.

Mihai Surdeanu is a research fellow in the Ramón y Cajal program of the Spanish Ministry of Science and Technology. Richard Johansson was funded by the Swedish National Graduate School of Language Technology (GSLT). Adam Meyers' participation was supported by the National Science Foundation, award CNS-0551615 (*Towards a Comprehensive Linguistic Annotation of Language*) and IIS-0534700 (*Collaborative Research: Structure Alignment-based Machine Translation*). Lluís Màrquez's participation was supported by the Spanish Ministry of Education and Science, through research projects Trangram (TIN2004-07925-C03-02) and OpenMT (TIN2006-15307-C03-02).

References

X. Carreras. 2007. Experiments with a Higher-Order Projective Dependency Parser. In *Proc. of CoNLL-2007 Shared Task.*

X. Carreras and L. Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proc. of CoNLL-2005.*

X. Carreras and L. Màrquez. 2004. Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In *Proc. of CoNLL-2004.*

W. Che, Z. Li, Y. Hu, Y. Li, B. Qin, T. Liu and S. Li. 2008. A Cascaded Syntactic and Semantic Dependency Parsing System. In *Proc. of CoNLL-2008 Shared Task.*

E. Chen, L. Shi and D. Hu. 2008. Probabilistic Model for Syntactic and Semantic Dependency Parsing. In *Proc. of CoNLL-2008 Shared Task.*

Chinchor, N. and P. Robinson. 1998. MUC-7 Named Entity Task Definition. In *Proc. of Seventh Message Understanding Conference (MUC-7).* http://www.itl.nist.gov/iaui/894.02/related_projects/ /muc/proceedings/muc_7_toc.html.

Chomsky, Noam. 1981. *Lectures on Government and Binding.* Foris Publications, Dordrecht.

Y.J. Chu and T.H. Liu. 1965. On the Shortest Arborescence of a Directed Graph. In *Science Sinica*, 14:1396-1400.

M. Ciaramita, G. Attardi, F. Dell'Orletta, and M. Surdeanu. 2008. DeSRL: A Linear-Time Semantic Role Labeling System. In *Proc. of CoNLL-2008 Shared Task.*

M. Ciaramita and Y. Altun. 2006. Broad Coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger. In *Proc. of EMNLP.*

M. Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania.

J. Edmonds. 1967. Optimum Branchings. In *Journal of Research of the National Bureau of Standards*, 71B:233–240.

J. Eisner. 2000. Bilexical Grammars and Their Cubic-Time Parsing Algorithms. *New Developments in Parsing Algorithms*, Kluwer Academic Publishers.

W. N. Francis and H. Kuĉera. 1964. Brown Corpus. Manual of Information to accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers. Revised 1971, Revised and Amplified 1979.

C. Fellbaum, editor. 1998. *WordNet: An electronic lexical database.* MIT Press.

J. Giménez and L. Màrquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proc. of LREC.*

K. Hacioglu. 2004. Semantic Role Labeling Using Dependency Trees. In *Proc. of COLING-2004.*

[12] http://code.google.com/p/whatswrong/

J. Henderson, P. Merlo, G. Musillo and I. Titov. 2008. A Latent Variable Model of Synchronous Parsing for Syntactic and Semantic Dependencies. In *Proc. of CoNLL-2008 Shared Task.*

R. Johansson and P. Nugues. 2008. Dependency-based Syntactic–Semantic Analysis with PropBank and NomBank. In *Proc. of CoNLL-2008 Shared Task.*

R. Johansson and P. Nugues. 2007. Extended Constituent-to-Dependency Conversion for English. In *Proc. of NODALIDA.*

X. Lluís and L. Màrquez. 2008. A Joint Model for Parsing Syntactic and Semantic Dependencies. In *Proc. of CoNLL-2008 Shared Task.*

D. Magerman. 1994. Natural Language Parsing as Statistical Pattern Recognition. Ph.D. thesis, Stanford University.

M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19.

R. McDonald, F. Pereira, K. Ribarov and J. Hajic. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms In *Proc. of HLT-EMNLP.*

A. Meyers, R. Grishman, M. Kosaka, and S. Zhao. 2001. Covering Treebanks with GLARF. In *Proc. of the ACL/EACL 2001 Workshop on Sharing Tools and Resources for Research and Education.*

Meyers, A., R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The NomBank Project: An Interim Report. In *NAACL/HLT 2004 Workshop Frontiers in Corpus Annotation*, Boston.

J. Nivre, J. Hall, J. Nilsson and G. Eryigit. 2006. Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines. In *Proc. of CoNLL-X Shared Task.*

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, D. Yuret. 2007. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proc. of CoNLL-2007.*

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryiğit, S. Kübler, S. Marinov, and E. Marsi. 2007b. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1).

S. Riedel and I. Meza-Ruiz. 2008. Collective Semantic Role Labelling with Markov Logic. In *Proc. of CoNLL-2008 Shared Task.*

Y. Samuelsson, O. Täckström, S. Velupillai, J. Eklund, M. Fishel and M. Saers. 2008. Mixing and Blending Syntactic and Semantic Dependencies. In *Proc. of CoNLL-2008 Shared Task.*

W. Sun, H. Li and Z. Sui. 2008. The Integration of Dependency Relation Classification and Semantic Role Labeling Using Bilayer Maximum Entropy Markov Models. In *Proc. of CoNLL-2008 Shared Task.*

E. F. Tjong Kim San and F. De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proc. of CoNLL-2003.*

D. Vickrey and D. Koller. 2008. Applying Sentence Simplification to the CoNLL-2008 Shared Task. In *Proc. of CoNLL-2008 Shared Task.*

R. Weischedel and A. Brunstein. 2005. BBN pronoun coreference and entity type corpus. Technical report, Linguistic Data Consortium.

H. Yamada and Y. Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In *Proc. of IWPT.*

Y. Zhang, R. Wang and H. Uszkoreit. 2008. Hybrid Learning of Dependency Structures from Heterogeneous Linguistic Resources. In *Proc. of CoNLL-2008 Shared Task.*

Zhao, S. 2005. *Information Extraction from Multiple Syntactic Sources.* Ph.D. thesis, NYU.