

# Relation Extraction: PPI task at BioCreAtIvE

Danielle Ben-Gera

IE in the Biomedical Domain  
Dr. Günter Neumann & Alejandro Pironti

# Outline

## BioCreAtIvE2: PPI Task

Motivation

The Task

Interaction Pairs Sub-task

Protein Interaction Method Sub-task

Protein Interaction Sentences Sub-task

Results

## AKANE system

The Architecture

Results

## Enju Parser

Overview

Stochastic Unification-based grammars

Experiment and Evaluation

# Outline

## BioCreAtIvE2: PPI Task

- Motivation

- The Task

  - Interaction Pairs Sub-task

  - Protein Interaction Method Sub-task

  - Protein Interaction Sentences Sub-task

- Results

## AKANE system

- The Architecture

- Results

## Enju Parser

- Overview

- Stochastic Unification-based grammars

- Experiment and Evaluation

# Outline

## BioCreAtIvE2: PPI Task

- Motivation

- The Task

  - Interaction Pairs Sub-task

  - Protein Interaction Method Sub-task

  - Protein Interaction Sentences Sub-task

- Results

## AKANE system

- The Architecture

- Results

## Enju Parser

- Overview

- Stochastic Unification-based grammars

- Experiment and Evaluation



# Motivation

- ▶ Development of new experimental methods makes it possible to study protein interaction at a much larger scale.
- ▶ Databases provide interaction information: IntAct, MINT.
- ▶ Manually curating protein interaction information is too slow
  - ⇒ Employ automated information extraction and text mining techniques.

# Motivation

- ▶ Development of new experimental methods makes it possible to study protein interaction at a much larger scale.
- ▶ Databases provide interaction information: IntAct, MINT.
- ▶ Manually curating protein interaction information is too slow
  - Employ automated information extraction and text mining techniques.

# Outline

## BioCreAtIvE2: PPI Task

Motivation

The Task

Interaction Pairs Sub-task

Protein Interaction Method Sub-task

Protein Interaction Sentences Sub-task

Results

## AKANE system

The Architecture

Results

## Enju Parser

Overview

Stochastic Unification-based grammars

Experiment and Evaluation



# PPI Task

- ▶ Determine the state of the art in extraction of PPI.
- ▶ Produce useful resources for training and testing PPI extraction systems.
- ▶ Learn which approaches are successful and practical.
- ▶ Provide the biology community with useful tools to extract PPI from text.

# The Data

- ▶ Training: 740 full text articles.
- ▶ Testing: 358 full text articles.
- ▶ Formats: PDF and HTML
  - ▶ Only full text articles which were available in both PDF and HTML formats were included.
  - ▶ Converted to simple text using `html2text` and `pdftotext`.
- ▶ No redundant articles were allowed.

## Interaction Pairs Sub-task (IPS)

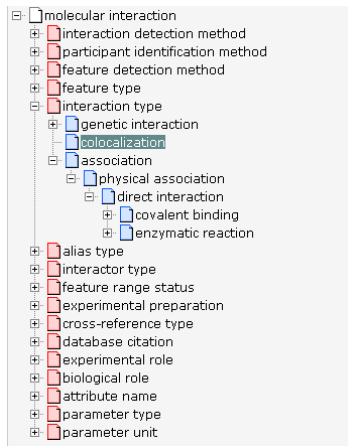
Identifying pairs of interacting proteins from full text articles.

- ▶ Normalized protein names  
(UniProt accession number)
  - ▶ Not all proteins are covered in UniProt.

# Interaction Pairs Sub-task (IPS)

Identifying pairs of interacting proteins from full text articles.

- ▶ Normalized protein names (UniProt accession number)
  - ▶ Not all proteins are covered in UniProt.
- ▶ *IntAct* and *MINT* curate all interaction that are:
  - ▶ interaction types
  - ▶ colocalisations
  - ▶ physical interactions



Molecular Interaction ontology  
via <http://www.ebi.ac.uk/ontology-lookup/>

# IPS: The setting

## Input: Training Data

- ▶ Full text articles.
- ▶ List with the interaction pairs that were curated from the articles and their synonyms.
- ▶ Test data: full articles.

## Output

- ▶ A ranked list of normalized protein-protein interaction pairs for each article.
- ▶ proteins must be disambiguated with respect to the organism source.

## Evaluation

- ▶ Precision, recall and F-measure.

# IPS: The setting

## Input: Training Data

- ▶ Full text articles.
- ▶ List with the interaction pairs that were curated from the articles and their synonyms.
- ▶ Test data: full articles.

## Output

- ▶ A ranked list of normalized protein-protein interaction pairs for each article.
- ▶ proteins must be disambiguated with respect to the organism source.

## Evaluation

- ▶ Precision, recall and F-measure.

# IPS: The setting

## Input: Training Data

- ▶ Full text articles.
- ▶ List with the interaction pairs that were curated from the articles and their synonyms.
- ▶ Test data: full articles.

## Output

- ▶ A ranked list of normalized protein-protein interaction pairs for each article.
- ▶ proteins must be disambiguated with respect to the organism source.

## Evaluation

- ▶ Precision, recall and F-measure.

# IPS: The setting

## Input: Training Data

- ▶ Full text articles.
- ▶ List with the interaction pairs that were curated from the articles and their synonyms.
- ▶ Test data: full articles.

## Output

- ▶ A ranked list of normalized protein-protein interaction pairs for each article.
- ▶ proteins must be disambiguated with respect to the organism source.

## Evaluation

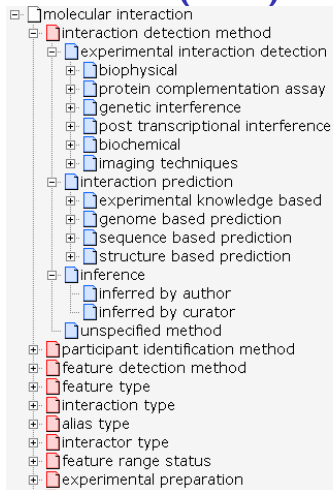
- ▶ Precision, recall and F-measure.



## Protein Interaction Method Sub-task (IMS)

From the text, extract the method(s) that were used to detect the interaction.

- ▶ Experimental methods are more reliable.
- ▶ MI ontology provides additional information: *definition, exact synonyms and related synonyms*



Molecular Interaction ontology  
via <http://www.ebi.ac.uk/ontology-lookup/>

# IMS: The setting

## Input: Training Data

- ▶ Full text articles corpus.
- ▶ List with the interaction pairs and their interaction detection methods.
- ▶ Test data: full articles, articles that describe large scale experiments were excluded

## Output

- ▶ A ranked list of normalized protein-protein interaction pairs for each article and their method.

## Evaluation

- ▶ Low number of submissions  $\Rightarrow$  measure only precision

# IMS: The setting

## Input: Training Data

- ▶ Full text articles corpus.
- ▶ List with the interaction pairs and their interaction detection methods.
- ▶ Test data: full articles, articles that describe large scale experiments were excluded

## Output

- ▶ A ranked list of normalized protein-protein interaction pairs for each article and their method.

## Evaluation

- ▶ Low number of submissions  $\Rightarrow$  measure only precision

# IMS: The setting

## Input: Training Data

- ▶ Full text articles corpus.
- ▶ List with the interaction pairs and their interaction detection methods.
- ▶ Test data: full articles, articles that describe large scale experiments were excluded

## Output

- ▶ A ranked list of normalized protein-protein interaction pairs for each article and their method.

## Evaluation

- ▶ Low number of submissions  $\Rightarrow$  measure only precision

# IMS: The setting

## Input: Training Data

- ▶ Full text articles corpus.
- ▶ List with the interaction pairs and their interaction detection methods.
- ▶ Test data: full articles, articles that describe large scale experiments were excluded

## Output

- ▶ A ranked list of normalized protein-protein interaction pairs for each article and their method.

## Evaluation

- ▶ Low number of submissions  $\Rightarrow$  measure only precision

# Protein Interaction Sentences Sub-task (ISS)

Extract supporting sentences: select the most relevant sentences expressing the interaction.

- ▶ A ranked list of maximum *five* evidence passages.
- ▶ each passage may contain up to *three* sentences.

## Evaluation

- ▶ Percentage of interaction-relevant sentences from the total number of sentences.
- ▶ Mean Reciprocal Rank (MRR): manually chosen best sentence.

# Protein Interaction Sentences Sub-task (ISS)

Extract supporting sentences: select the most relevant sentences expressing the interaction.

- ▶ A ranked list of maximum *five* evidence passages.
- ▶ each passage may contain up to *three* sentences.

## Evaluation

- ▶ Percentage of interaction-relevant sentences from the total number of sentences.
- ▶ Mean Reciprocal Rank (MRR): manually chosen best sentence.

# ISS: The Data

## Input: Training Data

- ▶ Limited amount of evidences were provided.
- ▶ Additional resources:
  - ▶ *Anne-Lise Veuthey corpus*: 697 sentences containing protein interactions (from PubMed).
  - ▶ *Prodisen interaction subset*: 921 sentences that are manually classified whether they contain an interaction description.
  - ▶ *Chirstine Brun corpus*: Sentences derived from abstracts with interactions and interaction types.
  - ▶ *GeneRIF interactions*: Collection of interaction sentences (51,381 entries)



# ISS: Trends

- ▶ Often used:
  - ▶ Sentence segmentation.
  - ▶ POS tagging.
  - ▶ Stemming.
  - ▶ Shallow parsing.
  - ▶ Regular expressions, pattern matching.
- ▶ Not used often:
  - ▶ Additional training data.
  - ▶ External lexical resources: dictionaries or ontologies.
  - ▶ Abbreviations extraction.
- ▶ Winning strategy:
  - ▶ Sophisticated protein mention and normalization techniques.
  - ▶ Species identification.

# ISS: Trends

- ▶ Often used:
  - ▶ Sentence segmentation.
  - ▶ POS tagging.
  - ▶ Stemming.
  - ▶ Shallow parsing.
  - ▶ Regular expressions, pattern matching.
- ▶ Not used often:
  - ▶ Additional training data.
  - ▶ External lexical resources: dictionaries or ontologies.
  - ▶ Abbreviations extraction.
- ▶ Winning strategy:
  - ▶ Sophisticated protein mention and normalization techniques.
  - ▶ Species identification.

# ISS: Trends

- ▶ Often used:
  - ▶ Sentence segmentation.
  - ▶ POS tagging.
  - ▶ Stemming.
  - ▶ Shallow parsing.
  - ▶ Regular expressions, pattern matching.
- ▶ Not used often:
  - ▶ Additional training data.
  - ▶ External lexical resources: dictionaries or ontologies.
  - ▶ Abbreviations extraction.
- ▶ Winning strategy:
  - ▶ Sophisticated protein mention and normalization techniques.
  - ▶ Species identification.

# Outline

## BioCreAtIvE2: PPI Task

Motivation

The Task

Interaction Pairs Sub-task

Protein Interaction Method Sub-task

Protein Interaction Sentences Sub-task

Results

## AKANE system

The Architecture

Results

## Enju Parser

Overview

Stochastic Unification-based grammars

Experiment and Evaluation

# IPS results

Average performance for an article in the test collection.

Two test collections:

- ▶ Whole test article collection.
- ▶ SwissProt-only articles set.

# IPS results

Average performance for an article in the test collection.

Two test collections:

- ▶ Whole test article collection.
- ▶ SwissProt-only articles set.

Team	Precision	Recall	F-score
4(1)	0.3893	0.3073	0.2885
6(1)	0.2758	0.3011	0.2532
6(2)	0.2218	0.2592	0.2066
6(3)	0.2392	0.3025	0.2272
14(1)	0.1791	0.1421	0.1384
14(2)	0.1944	0.1300	0.1414
14(3)	0.1162	0.1057	0.0985
28(1)	0.1373	0.2905	0.1579
28(2)	0.2177	0.2651	0.2039
28(3)	0.3096	0.2935	0.2623
36(1)	0.0441	0.1121	0.0503
36(2)	0.0229	0.0990	0.0305
36(3)	0.0548	0.1350	0.0680
40(1)	0.0762	0.2489	0.0990
40(2)	0.2632	0.2484	0.2172
58(1)	0.0003	0.0006	0.0004

# IPS results

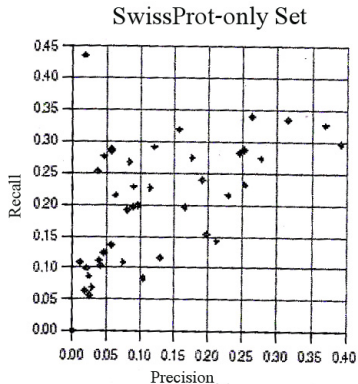
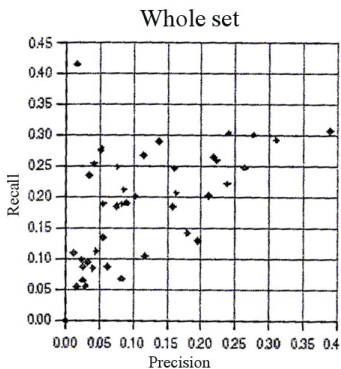
Average performance for an article in the test collection.

Two test collections:

- ▶ Whole test article collection.
- ▶ SwissProt-only articles set.

Team	Precision	Recall	F-score
4(1)	0.3908	0.2970	0.2849
6(1)	0.3150	0.3356	0.2871
6(2)	0.2519	0.2868	0.2308
6(3)	0.2632	0.3394	0.2532
14(1)	0.1975	0.1543	0.1510
14(2)	0.2133	0.1430	0.1552
14(3)	0.1287	0.1157	0.1079
28(1)	0.1373	0.2905	0.1579
28(2)	0.2177	0.2651	0.2039
28(3)	0.3096	0.2935	0.2623
36(1)	0.0456	0.1243	0.0560
36(2)	0.0202	0.0997	0.0295
36(3)	0.0560	0.1362	0.0686
40(1)	0.0824	0.2672	0.1083
40(2)	0.2751	0.2737	0.2355
58(1)	0.0000	0.0000	0.0000

# IPS Results: Whole set vs. SwissProt-only





## IPS Results: Normalization evaluation

The normalization of the protein names were also evaluated

- ▶ Precision:
  - ▶ Team 4: 0.56 (whole set)
  - ▶ Team 28: 0.57 (SwissProt-only)
- ▶ Recall:
  - ▶ Team 42: 0.68 (whole set)
  - ▶ Team 42: 0.69<sup>1</sup> (SwissProt-only)
- ▶ F-score:
  - ▶ Team 28: 0.52 (whole set)
  - ▶ Team 4: 0.48 (SwissProt-only)

---

<sup>1</sup>But very low Recall: 0.08

# IMS results

▣ Only 2 teams submitted systems.

**Evaluation Method:** Using a manually annotated gold standard.  
(Annotated using the Molecular Interaction controlled vocabulary)

- ▶ Exact Matching.
- ▶ Parent Matching.

# IMS results

▣ Only 2 teams submitted systems.

## Exact Match Results:

Team	Precision	Recall	F-score
14(1)	0.3628	0.2172	0.2513
14(2)	0.3186	0.1980	0.2249
14(3)	0.3348	0.1938	0.2265
40(3)	<b>0.6679</b>	0.3383	0.4207
40(1)	0.4028	<b>0.5548</b>	0.4363
40(2)	0.5068	0.5222	<b>0.4836</b>

## Parent Matching Results:

Team	Precision	Recall	F-score
14(1)	0.4986	0.3078	0.3495
14(2)	0.4471	0.2847	0.3170
14(3)	0.4881	0.2953	0.3375
40(3)	<b>0.6794</b>	0.3472	0.4302
40(1)	0.5899	<b>0.8548</b>	<b>0.6519</b>
40(2)	0.6541	0.7093	0.6375

## ISS Results: Evaluation Method

- ▶ Compare passages to manually extracted passages.
- ▶ Sliding the shorter one over the other and calculate string similarity for each position.
- ▶ Correct if string similarity is high.
- ▶ Problems:
  - ▶ Passage might be correct but was not chosen by the curators.
  - ▶ For some articles curators did not find a suitable evidence sentence.

## ISS Results: Evaluation Method

- ▶ Compare passages to manually extracted passages.
- ▶ Sliding the shorter one over the other and calculate string similarity for each position.
- ▶ Correct if string similarity is high.
- ▶ Problems:
  - ▶ Passage might be correct but was not chosen by the curators.
  - ▶ For some articles curators did not find a suitable evidence sentence.

# ISS Results

Team	Precision	Precision (unique)	MRR
4(1)	0.1371	0.1413	-
4(2)	<b>0.1909</b>	<b>0.1939</b>	-
6(1)	0.0589	0.0565	0.5525
14(1)	0.0205	0.0207	<b>0.8718</b>
14(2)	0.0218	0.0237	0.8167
14(3)	0.0232	0.0214	<b>0.8718</b>
28(1)	0.0495	0.0493	0.3740
28(2)	0.0565	0.0565	0.3696
28(3)	0.0646	0.0595	0.3392
36(1)	0.0514	0.0496	0.5731
36(2)	0.0483	0.0456	0.5813
36(3)	0.0605	0.0533	0.5476

## Discussion and Conclusions

Common difficulties:

- ▶ Errors resulting from converting PDF to HTML.
- ▶ Processing tables, figures, legends etc.
- ▶ Imperfect protein names normalization step.
- ▶ Heavy use of domain specific terminology.
- ▶ Errors in NLP components (e.g. POS-tagger, shallow parsing) that were trained on an open domain English corpus

Observations:

- ▶ More sophisticated approaches outperformed simple ones.
- ▶ Gene name recognition and normalization is very important.

# Outline

## BioCreAtIvE2: PPI Task

Motivation

The Task

Interaction Pairs Sub-task

Protein Interaction Method Sub-task

Protein Interaction Sentences Sub-task

Results

## AKANE system

The Architecture

Results

## Enju Parser

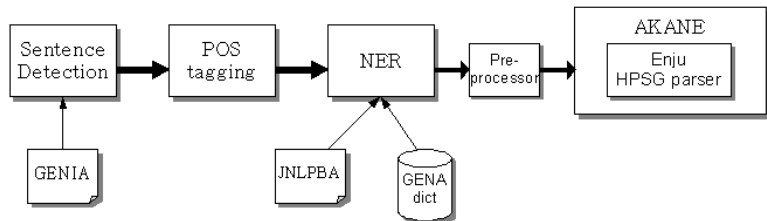
Overview

Stochastic Unification-based grammars

Experiment and Evaluation



## The Architecture: Overview



- ▶ All modules use machine learning.

# Sentence Detecting

- ▶ Method: Maximum Entropy.
- ▶ Training: GENIA corpus.
- ▶ Candidates: positions for splitting using selected delimiters.
- ▶ Features: delimiters, prev/next words, special characters, numbers, case.
- ▶ F-score: 99.7, but more errors when tested on BioCreAtIvE (due to HTML encoding and figure explanations texts).

# Name Entity Recognition

- ▶ Step 1: [statistical recognizer](#)
  - ▶ Training: JNLPBA 2004 data.
  - ▶ Outputs: The probability that a substring of a sentence is a protein name.
- ▶ Step 2: [Mapping to a Dictionary](#)
  - ▶ Candidates: strings from Step 1 that are above a threshold.
  - ▶ Outputs: set of strings that could be mapped to entries in the dictionary with a given edit distance.
- ▶ Step 3: [Disambiguating](#)
  - ▶ Method: Maximum Entropy.
  - ▶ Training: 266 articles from the data.
  - ▶ Features: similarity between article and MEDLINE articles (referred by Uniprot), source dictionary (Uniprot/GENA), edit distance, type (protein name, gene name,...)

# Name Entity Recognition

- ▶ Step 1: [statistical recognizer](#)
  - ▶ Training: JNLPBA 2004 data.
  - ▶ Outputs: The probability that a substring of a sentence is a protein name.
- ▶ Step 2: [Mapping to a Dictionary](#)
  - ▶ Candidates: strings from Step 1 that are above a threshold.
  - ▶ Outputs: set of strings that could be mapped to entries in the dictionary with a given edit distance.
- ▶ Step 3: [Disambiguating](#)
  - ▶ Method: Maximum Entropy.
  - ▶ Training: 266 articles from the data.
  - ▶ Features: similarity between article and MEDLINE articles (referred by Uniprot), source dictionary (Uniprot/GENA), edit distance, type (protein name, gene name,...)

# Name Entity Recognition

- ▶ Step 1: [statistical recognizer](#)
  - ▶ Training: JNLPBA 2004 data.
  - ▶ Outputs: The probability that a substring of a sentence is a protein name.
- ▶ Step 2: [Mapping to a Dictionary](#)
  - ▶ Candidates: strings from Step 1 that are above a threshold.
  - ▶ Outputs: set of strings that could be mapped to entries in the dictionary with a given edit distance.
- ▶ Step 3: [Disambiguating](#)
  - ▶ Method: Maximum Entropy.
  - ▶ Training: 266 articles from the data.
  - ▶ Features: similarity between article and MEDLINE articles (referred by Uniprot), source dictionary (Uniprot/GENA), edit distance, type (protein name, gene name,..)

## AKANE system

- ▶ AKANE system requires Almed corpus style input
  - ➡ use preprocessor to automatically create this style.
- ▶ Parse input text using *Enju* HPSG parser.
  - ▶ Trained with Penn Treebank
  - ▶ Domain adaption<sup>2</sup>: using GENIA Treebank
  - ▶ Bio-performance: 86.9 F-score
- ▶ Creates the smallest possible parse tree that covers both proteins.
- ▶ Postprocessor:
  - ▶ pick only the most likely interaction pair based on NER probabilities.
  - ▶ Filter out results where proteins did not have identical species tags.

---

<sup>2</sup>[Hara et al, 2005]

## AKANE system

- ▶ AKANE system requires Almed corpus style input
  - ➡ use preprocessor to automatically create this style.
- ▶ Parse input text using *Enju* HPSG parser.
  - ▶ Trained with Penn Treebank
  - ▶ Domain adaption<sup>2</sup>: using GENIA Treebank
  - ▶ Bio-performance: 86.9 F-score
- ▶ Creates the smallest possible parse tree that covers both proteins.
- ▶ Postprocessor:
  - ▶ pick only the most likely interaction pair based on NER probabilities.
  - ▶ Filter out results where proteins did not have identical species tags.

---

<sup>2</sup>[Hara et al, 2005]

## AKANE system

- ▶ AKANE system requires Almed corpus style input
  - ➡ use preprocessor to automatically create this style.
- ▶ Parse input text using *Enju* HPSG parser.
  - ▶ Trained with Penn Treebank
  - ▶ Domain adaption<sup>2</sup>: using GENIA Treebank
  - ▶ Bio-performance: 86.9 F-score
- ▶ Creates the smallest possible parse tree that covers both proteins.
- ▶ Postprocessor:
  - ▶ pick only the most likely interaction pair based on NER probabilities.
  - ▶ Filter out results where proteins did not have identical species tags.

---

<sup>2</sup>[Hara et al, 2005]



## AKANE system

- ▶ AKANE system requires Almed corpus style input
  - ➡ use preprocessor to automatically create this style.
- ▶ Parse input text using *Enju* HPSG parser.
  - ▶ Trained with Penn Treebank
  - ▶ Domain adaption<sup>2</sup>: using GENIA Treebank
  - ▶ Bio-performance: 86.9 F-score
- ▶ Creates the smallest possible parse tree that covers both proteins.
- ▶ Postprocessor:
  - ▶ pick only the most likely interaction pair based on NER probabilities.
  - ▶ Filter out results where proteins did not have identical species tags.

---

<sup>2</sup>[Hara et al, 2005]

# Outline

## BioCreAtIvE2: PPI Task

Motivation

The Task

Interaction Pairs Sub-task

Protein Interaction Method Sub-task

Protein Interaction Sentences Sub-task

Results

## AKANE system

The Architecture

Results

## Enju Parser

Overview

Stochastic Unification-based grammars

Experiment and Evaluation

# Three Runs

## Run 1

- ▶ Not using inter-species interaction filter.
  - ▶ P: 8.2%, R:14.6%, F-score: 10.5.

## Run 2

- ▶ Best run in terms of F-score on training set.
  - ▶ P: 10.6%, R:19.1%, F-score: 13.7.

## Run 3

- ▶ Best run in terms of F-score on Almed set.
  - ▶ P: 15.7%, R:15.9%, F-score: 15.8.

⇒ Training on full text did not preform better than training on Almed abstracts.

# Three Runs

## Run 1

- ▶ Not using inter-species interaction filter.
  - ▶ P: 8.2%, R:14.6%, F-score: 10.5.

## Run 2

- ▶ Best run in terms of F-score on training set.
  - ▶ P: 10.6%, R:19.1%, F-score: 13.7.

## Run 3

- ▶ Best run in terms of F-score on Almed set.
  - ▶ P: 15.7%, R:15.9%, F-score: 15.8.

⇒ Training on full text did not preform better than training on Almed abstracts.

## Three Runs

### Run 1

- ▶ Not using inter-species interaction filter.
  - ▶ P: 8.2%, R:14.6%, F-score: 10.5.

### Run 2

- ▶ Best run in terms of F-score on training set.
  - ▶ P: 10.6%, R:19.1%, F-score: 13.7.

### Run 3

- ▶ Best run in terms of F-score on Almed set.
  - ▶ P: 15.7%, R:15.9%, F-score: 15.8.

▮ Training on full text did not preform better than training on Almed abstracts.

## Three Runs

### Run 1

- ▶ Not using inter-species interaction filter.
  - ▶ P: 8.2%, R:14.6%, F-score: 10.5.

### Run 2

- ▶ Best run in terms of F-score on training set.
  - ▶ P: 10.6%, R:19.1%, F-score: 13.7.

### Run 3

- ▶ Best run in terms of F-score on Almed set.
  - ▶ P: 15.7%, R:15.9%, F-score: 15.8.

▣ Training on full text did not preform better than training on Almed abstracts.

## Discussion

- ▶ Underlying assumption: All sentences that include the mentioning of two proteins that were marked as interacting by the BioCreAtIvE training data, were describing the interaction.
- ▶ Only 250 out 740 training articles were used since AKANE did not scale well to large amount of text. (originally build for Almed abstracts).
  - ▶ Use only articles where all interacting protein names/IDs could be recognized by the NER.
  - ▶ Discard articles with too many co-occurrence sentences

# Outline

## BioCreAtIvE2: PPI Task

Motivation

The Task

Interaction Pairs Sub-task

Protein Interaction Method Sub-task

Protein Interaction Sentences Sub-task

Results

## AKANE system

The Architecture

Results

## Enju Parser

Overview

Stochastic Unification-based grammars

Experiment and Evaluation



# Charecharistics and Motivation

- ▶ Unification-based grammars give in-depth useful analysis.  
But: computing all possible parses is very expensive.
- ▶ Using probabilistic models of Unification-based grammars allows us to employ techniques from PCFG parsing:
  - ▶ Viterbi parsing.
  - ▶ Local (beam) thresholding.
  - ▶ Global (beam) thresholding.
  - ▶ Quick check.
  - ▶ Large Constituent inhibition.
  - ▶ Chunk parsing.

## Charecharistics and Motivation

- ▶ Unification-based grammars give in-depth useful analysis.  
But: computing all possible parses is very expensive.
- ▶ Using probabilistic models of Unification-based grammars allows us to employ techniques from PCFG parsing:
  - ▶ Viterbi parsing.
  - ▶ Local (beam) thresholding.
  - ▶ Global (beam) thresholding.
  - ▶ Quick check.
  - ▶ Large Constituent inhibition.
  - ▶ Chunk parsing.

# Outline

## BioCreAtIvE2: PPI Task

Motivation

The Task

Interaction Pairs Sub-task

Protein Interaction Method Sub-task

Protein Interaction Sentences Sub-task

Results

## AKANE system

The Architecture

Results

## Enju Parser

Overview

Stochastic Unification-based grammars

Experiment and Evaluation

## hpsg and Probabilistic models

### Maximum Entropy model (discrete)<sup>3</sup>

$$p(T|\omega) = \frac{1}{Z_\omega} \exp\left(\sum_u \lambda_u f_u(T)\right)$$

$$Z: \text{normalizes the probability} = \sum_{T'} \exp\left(\sum_u \lambda_u f_u(T)\right)$$

$\lambda_u$ : the model parameter, estimated to maximize likelihood.

$f_u(T)$ : the feature function - frequency of  $f_u$  in  $T$ .

In practice, the model is hard to estimate:

- ▶ requires computation of  $p(T|\omega)$  for every possible  $T$ .
- ▶ simplify computation by assuming  $T = \{c_1, \dots, c_n\}$  and the set of all parse trees is represented as a AND/OR graph.

---

<sup>3</sup>[Abney, 1997]

## hpsg and Probabilistic models

### Maximum Entropy model (discrete)<sup>3</sup>

$$p(T|\omega) = \frac{1}{Z_\omega} \exp \left( \sum_u \lambda_u f_u(T) \right)$$

$Z$ : normalizes the probability =  $\sum_{T'} \exp \left( \sum_u \lambda_u f_u(T) \right)$

$\lambda_u$ : the model parameter, estimated to maximize likelihood.

$f_u(T)$ : the feature function - frequency of  $f_u$  in  $T$ .

In practice, the model is hard to estimate:

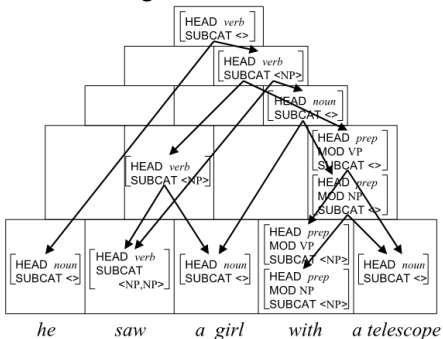
- ▶ requires computation of  $p(T|\omega)$  for every possible  $T$ .
- ▶ simplify computation by assuming  $T = \{c_1, \dots, c_n\}$  and the set of all parse trees is represented as a AND/OR graph.

---

<sup>3</sup>[Abney, 1997]

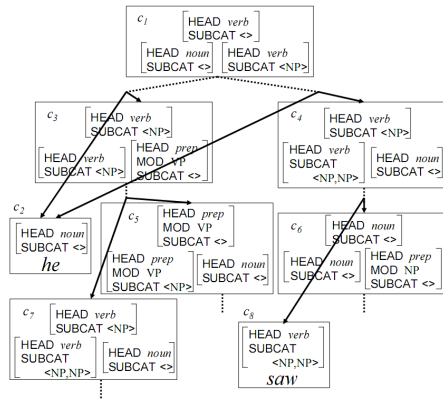
# HPSG and Probabilistic models

## Chart Parsing



Figures are taken from [Hara et al, 2005]

## Packed representation



## Probabilistic models: Packed Representation

Using packed representation we can rewrite the probability of a parse:

$$p(T|\omega) = \frac{1}{Z_\omega} \exp \left( \sum_{c \in T} \sum_u \lambda_u f_u(c) \right)$$

We can now use features for a node  $c$

---

RULE	the name of the applied schema
DIST	the distance between the head words of the daughters
COMMA	whether a comma exists between daughters and/or inside of daughter phrases
SPAN	the number of words dominated by the phrase
SYM	the symbol of the phrasal category (e.g. NP, VP)
WORD	the surface form of the head word
POS	the part-of-speech of the head word
LE	the lexical entry assigned to the head word

---

## Probabilistic models: Features

Each type of node has a different set of features:

$$f_{\text{binary}} = \left\langle \begin{array}{l} \text{RULE, DIST, COMMA,} \\ \text{SPAN}_l, \text{SYM}_l, \text{WORD}_l, \text{POS}_l, \text{LE}_l, \\ \text{SPAN}_r, \text{SYM}_r, \text{WORD}_r, \text{POS}_r, \text{LE}_r \end{array} \right\rangle$$

$$f_{\text{unary}} = \langle \text{RULE, SYM, WORD, POS, LE} \rangle$$

$$f_{\text{root}} = \langle \text{SYM, WORD, POS, LE} \rangle$$

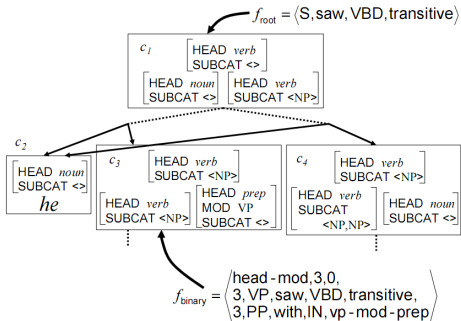


Figure is taken from [Hara et al, 2005]



# Improving Parsing Efficiency Techniques

- ▶ Quick check:
  - ▶ filters out non-unifiable structures by peeping
- ▶ Large constituent inhibition:
  - ▶ prevents the parser from generating medial edges that span more than k words length.
- ▶ CFG chunk parser: parse trees without non-terminal symbols.
  - ▶ Use bracket information to prevent the parser from generating cross brackets edges.
  - ▶ Chunk parsing is fast and precise.
  - ▶ CFG grammar is automatically extracted while generating the HPSG grammar (from Penn treebank)

# Improving Parsing Efficiency Techniques

- ▶ Quick check:
  - ▶ filters out non-unifiable structures by peeping
- ▶ Large constituent inhibition:
  - ▶ prevents the parser from generating medial edges that span more than k words length.
- ▶ CFG chunk parser: parse trees without non-terminal symbols.
  - ▶ Use bracket information to prevent the parser from generating cross brackets edges.
  - ▶ Chunk parsing is fast and precise.
  - ▶ CFG grammar is automatically extracted while generating the HPSG grammar (from Penn treebank)

# Improving Parsing Efficiency Techniques

- ▶ Quick check:
  - ▶ filters out non-unifiable structures by peeping
- ▶ Large constituent inhibition:
  - ▶ prevents the parser from generating medial edges that span more than  $k$  words length.
- ▶ CFG chunk parser: parse trees without non-terminal symbols.
  - ▶ Use bracket information to prevent the parser from generating cross brackets edges.
  - ▶ Chunk parsing is fast and precise.
  - ▶ CFG grammar is automatically extracted while generating the HPSG grammar (from Penn treebank)

## Chunk parsing

							0,7
						0,6	1,7
				0,5		1,6	2,7
			0,4		1,5	2,6	3,7
		0,3	1,4		2,5	3,6	4,7
	0,2	1,3	2,4		3,5	4,6	5,7
0,1	1,2	2,3	3,4	4,5	0,6		6,7
0	He	1	Runs	2	[	a	3
			Big	4	Company	5	]
					Very	6	Well
							7

- ▶ The HPSG parser never generates edges corresponding to the gray cells.

## Beam Thresholding

- Simple and effective technique for efficient parsing.
  - ▶ Local thresholding by number of edges
    - ▶ Keep top  $k$  edges.
    - ▶ Calculated at the end of the computation for each cell.
  - ▶ Local thresholding by beam width
    - ▶ Keep all edges with FOM greater than  $\alpha_{max} - \delta$ .
    - ▶  $\alpha_{max}$ : highest FOM among the edges of the cell.
    - ▶ Calculated at the end of the computation for each cell.
  - ▶ Global thresholding by beam width
    - ▶ Global FOM is greater than  $\alpha_{max} - \theta$   $\alpha_{max}$ : highest global FOM in the chart.
    - ▶ Global FOM of an edge is its FOM plus the outside FOM
  - ▶ Iterative beam thresholding
    - ▶ Start with a narrow beam and decrease the threshold if the parser did not yield a result.

## Beam Thresholding

- ▣ Simple and effective technique for efficient parsing.
  - ▶ Local thresholding by number of edges
    - ▶ Keep top  $k$  edges.
    - ▶ Calculated at the end of the computation for each cell.
  - ▶ Local thresholding by beam width
    - ▶ Keep all edges with FOM greater than  $\alpha_{max} - \delta$ .
    - ▶  $\alpha_{max}$ : highest FOM among the edges of the cell.
    - ▶ Calculated at the end of the computation for each cell.
  - ▶ Global thresholding by beam width
    - ▶ Global FOM is greater than  $\alpha_{max} - \theta$   $\alpha_{max}$ : highest global FOM in the chart.
    - ▶ Global FOM of an edge is its FOM plus the outside FOM
  - ▶ Iterative beam thresholding
    - ▶ Start with a narrow beam and decrease the threshold if the parser did not yield a result.

## Beam Thresholding

- Simple and effective technique for efficient parsing.
  - ▶ Local thresholding by number of edges
    - ▶ Keep top  $k$  edges.
    - ▶ Calculated at the end of the computation for each cell.
  - ▶ Local thresholding by beam width
    - ▶ Keep all edges with FOM greater than  $\alpha_{max} - \delta$ .
    - $\alpha_{max}$ : highest FOM among the edges of the cell.
    - ▶ Calculated at the end of the computation for each cell.
  - ▶ Global thresholding by beam width
    - ▶ Global FOM is greater than  $\alpha_{max} - \theta \alpha_{max}$ : highest global FOM in the chart.
    - ▶ Global FOM of an edge is its FOM plus the outside FOM
  - ▶ Iterative beam thresholding
    - ▶ Start with a narrow beam and decrease the threshold if the parser did not yield a result.

## Beam Thresholding

- Simple and effective technique for efficient parsing.
  - ▶ Local thresholding by number of edges
    - ▶ Keep top  $k$  edges.
    - ▶ Calculated at the end of the computation for each cell.
  - ▶ Local thresholding by beam width
    - ▶ Keep all edges with FOM greater than  $\alpha_{max} - \delta$ .  
 $\alpha_{max}$ : highest FOM among the edges of the cell.
    - ▶ Calculated at the end of the computation for each cell.
  - ▶ Global thresholding by beam width
    - ▶ Global FOM is greater than  $\alpha_{max} - \theta \alpha_{max}$ : highest global FOM in the chart.
    - ▶ Global FOM of an edge is its FOM plus the outside FOM
  - ▶ Iterative beam thresholding
    - ▶ Start with a narrow beam and decrease the threshold if the parser did not yield a result.



# Outline

## BioCreAtIvE2: PPI Task

Motivation

The Task

Interaction Pairs Sub-task

Protein Interaction Method Sub-task

Protein Interaction Sentences Sub-task

Results

## AKANE system

The Architecture

Results

## Enju Parser

Overview

Stochastic Unification-based grammars

Experiment and Evaluation

# The Gramamr and the model

- ▶ 2,284 Lexical Entries templates (plus lexical entries for POS, which were assigned to unknown words).
- ▶ 529,856 features in the model (trained on section 02-21).
- ▶ The thresholding parameters (determined manually from section 22):
  - ▶  $k_0 = 6.0$ ,  $\Delta k = 3.0$ ,  $k_{last} = 15.1$
  - ▶  $\delta_0 = 12$ ,  $\Delta\delta = 6$ ,  $k\delta = 30$
  - ▶  $\theta_0 = 8.0$ ,  $\Delta\theta = 4.0$ ,  $k\theta = 20.1$

## Results: Tested on Penn

Experimental results for development set (section 22) and test set (section 23) with manually annotated POS

	Precision	Recall	F-score	Avg. Time (ms)	No. of failed sentences
development set ( $\leq 40$ )	88.15%	87.34%	87.74%	352	11
development set ( $\leq 100$ )	87.57%	86.31%	86.94%	448	15
test set ( $\leq 40$ )	87.70%	86.78%	87.23%	355	14
test set ( $\leq 100$ )	87.26%	86.18%	86.71%	448	17

Experimental results for development set (section 22) and test set (section 23) with a POS tagger

	Precision	Recall	F-score	Avg. Time (ms)	No. of failed sentences
development set ( $\leq 40$ )	85.92%	85.00%	85.45%	364	8
development set ( $\leq 100$ )	85.31%	84.31%	84.81%	493	9
test set ( $\leq 40$ )	85.38%	84.65%	85.01%	397	15
test set ( $\leq 100$ )	84.99%	83.99%	84.49%	517	19

# Results

Viterbi alone vs. beam thresholding vs. iterative thresholding

	Precision	Recall	F-score	Avg. Time (ms)	No. of failed sentences
viterbi (none)	87.80%	87.52%	87.66%	94374	2
beam search (num+width)	88.63%	82.45%	85.43%	90	25
iterative (iterative)	87.40%	87.03%	87.21%	101	2

# Results

Performance for the sentences in Section 24 of  $\leq 40$  words

	Precision	Recall	F-score	Avg. Time (ms)	diff(*)	No. of failed sentences
full	85.35%	84.10%	84.72%	419	0	13
full-piter	85.47%	84.70%	85.09%	639	220	8
full-qc	85.35%	84.10%	84.72%	566	147	13
full-chp	85.65%	84.87%	85.26%	527	108	8
full-global	85.35%	84.33%	84.84%	457	38	10
full-lci	85.57%	84.33%	84.95%	439	20	13
full-piter-qc-chp-global-lci	85.49%	84.76%	85.12%	1100	681	7

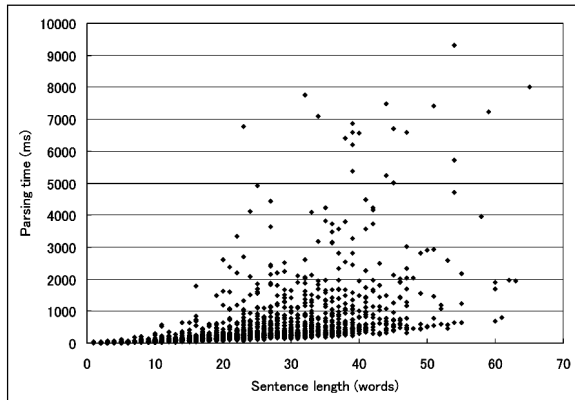
Performance for the sentences in Section 24 of  $\leq 100$  words

	Precision	Recall	F-score	Avg. Time (ms)	diff(*)	No. of failed sentences
full	85.07%	83.61%	84.34%	525	0	15
full-piter	85.20%	84.16%	84.68%	877	352	10
full-qc	85.07%	83.61%	84.34%	728	203	15
full-chp	85.34%	84.30%	84.81%	690	165	10
full-global	84.88%	84.02%	84.45%	545	20	10
full-lci	85.32%	83.85%	84.58%	576	51	15
full-piter-qc-chp-global-lci	85.17%	84.09%	84.63%	1591	1066	10

full	...	iterative + global + chp
piter	...	preserved iterative parsing
qc	...	quick check
chp	...	parsing with CFG chunk parser
lci	...	large constituent inhibition
diff(*)	...	(Avg. Time of full) - (Avg. Time)
full- $X_1, \dots, X_n$	...	$X_1, \dots, X_n$ are deactivated

# Results

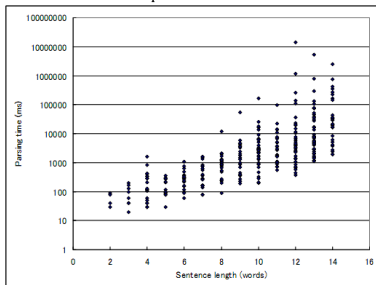
Parsing time versus sentence length ( $\leq 100$ )



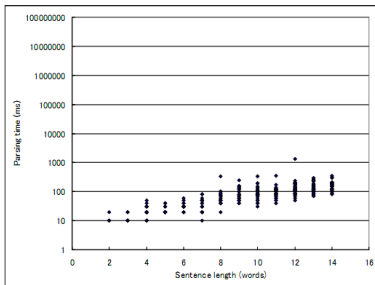
# Results

Parsing time versus sentence length ( $\leq 14$ ):

Simple Viterbi



Iterative Beam Thresholding



## Resources and References

- ▶ M. Krallinger, F. Leitner and A. Valencia. *Assessment of the Second BioCreative PPI task: Automatic Extraction of Protein-Protein Interactions*.
- ▶ Ontology Lookup Service: <http://www.ebi.ac.uk/ontology-lookup/>
- ▶ R. Sætre, K. Yoshida, A. Yakushiji, Y. Miyao, Y. Matsubayashi and T. Ohta, *AKANE System: Protein-Protein Interaction Pairs in the BioCreAtIvE2 Challenge, PPI-IPS subtask*. In the Proceedings of the Second BioCreative Challenge Evaluation Workshop. Spain. April, 2007.
- ▶ GENIA Treebank: <http://www-tsujii.is.s.u-tokyo.ac.jp/genia/topics/Corpus/GTB.html>
- ▶ BioNLP: <http://www.bionlp.org/>
- ▶ T. Hara, Y. Miyao and J. Tsujii, *Adapting a probabilistic disambiguation model of an HPSG parser to a new domain*, in IJCNLP 2005, vol. 3651 of LNAI, 199-210. Springer-Verlag. Korea. Oct, 2005.
- ▶ T. Ninomiya, Y. Tsuruoka, Y. Miyao, K. Taura, and J. Tsujii, *Fast and Scalable HPSG Parsing*. Traitement automatique des langues (TAL), 46(2). 2006.
- ▶ Y. Miyao and J. Tsujii, *Probabilistic disambiguation models for wide-coverage HPSG parsing*. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. 2005.
- ▶ S. Geman and M. Johnson, *Dynamic Programming for Parsing and Estimation of Stochastic Unification-Based Grammars*. In proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002