Improving Text-mining PPIs by Learning to Filter Entities from the Output of a Knowledge Extractor

Michael Gabilondo

Motivation

- A rule-based system has already been developed to extract PPIs from AIMed with high precision (89%, but ~96-97% throwing out inter-annotator disagreements) and modest recall (~68-70%)
- Verb predicates in an ontology have been defined for the domain; a predicate represents a meaning of a verb and specifies its arguments in terms of syntactic/semantic restrictions
- The semantic interpreter (SI) uses verb predicates and a noun ontology to link clauses to predicates and label their arguments with semantic roles
- The knowledge extractor (KE) uses (1) semantic extraction rules to extract the desired relations/arguments from the SI, and then (2) syntactic/semantic filtering rules to extract the desired entities from those arguments
- The extraction and filtering rules will vary depending on what is to be extracted – as determined by a gold-standard or human judge
- Developing the filtering rules for AIMed took time – it was not always clear why something did or didn't interact, and there seem to be inconsistencies and errors within the annotation
- The goal is to ease the burden of applying the system to a new domain using a gold-standard – by automatically learning to filter entities from arguments in the final step

Related Work – Rule-based methods – Mostly manual rules for extraction from parse trees – and a custom chunker for relation extraction (Šarić, 2004)

- **“Event Extraction from Biomedical Papers Using a Full Parser”, 2001, Yakushiji et. al.**
  - Define mapping rules from argument structures (containing grammatical relations) to frame structures (containing slots/semantic roles) – Bad PP attachment was a major problem – *No evaluation of the system on an information extraction task*
- **“Extracting regulatory gene expression networks from PubMed”, 2004, Šarić, et. al.**
  - CASS grammar, NP and relation chunker – e.g., up- or down- regulation of gene expression, e.g., prevent formation of complex – High precision but no evaluation on one of five the corpora
- **“IntEx: A Syntactic Role Driven Protein-Protein Interaction Extractor for Bio-Medical Text”, 2005, Ahmed et. al –**
  - Splits parse tree into clauses and uses a general algorithm for extracting PPIs based on a list of interaction terms – **65.66 % precision, 26.94% recall on DIP abstracts** – 45% of the errors were caused by bad protein tagging – (Low recall caused by ignoring full-text articles ...)
- **“Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach”, 2007, Rinaldi et. Al**
  - Full/deep dependency-based parsing; **GENIA corpus for evaluation; low/mid 90s precision**; also extracts “patterns describing biological interactions”
- **“RelEx–Relation extraction using dependency parse trees”, 2007, Fundel et. al.**
  - NER with a dictionary, interaction terms, parsing and some manual rules to identify relations – **p = 79, r = 85 on LLL**
- **“High precision rule based PPI extraction and per-pair basis performance evaluation”, 2012, Lee et. al.**
  - A few simple rules on the output of the parser; AImed evaluation only – **94-97% precision, 15-24% recall** (using our per-instance evaluation)

Related Work – ML-based methods – SVM classification of feature vectors – PPI features extracted from parse

- **“Syntactic features for protein-protein interaction extraction”, 2007, Saetre et. al.**
  - Enju parser is used to generate feature vectors for SVM PPI classification using tree kernel – Enju is a deep parser that identifies arguments of verbs by linking them to a syntactic predicate-argument structures like verb_arg12 (ARG_i labeling scheme)
  - **AIMed 10-F CV: 64.3% precision, 44.1% recall, 52.0% F-score**
- **“Extracting Protein Interactions from Text with the Unified AkaneRE Event Extraction System”, Saetre et. Al, 2010.**
  - For event extraction, extracted 9 generalized templates from GENIA, which contain syntactic and semantic information about arguments (# of arguments, semantic roles, NE type of each role) – Learn which ne/event combinations go in each template, based on training data – But placed 6th in BioNLP-EE
  - **AIMed 10-F CV 62.7% precision, 66.6% recall, 64.2% F-score**
- **“A hybrid approach to extract protein–protein interactions”, Bui et. Al, 2011.**
  - Extract candidate PPI pairs from the parse tree using high-recall manual rules, and then test/train a classifier on the output to produce a final classification (SVM, features from parse tree, paths between proteins)
  - **AIMed 10-F CV: 55.3% precision, 68.5% recall, 61.2% F-score**

Motivation for Predicates – Syntactic variations of "interaction" in AIMed

- "Our data suggest that **TR6** inhibits the ***interactions of*** **LIGHT** ***with*** **HVEM**/**TR2** and **LTbetaR** , thereby suppressing **LIGHT**-mediated HT29 cell death."
- "**Presenilin 1** suppresses the function of **c-Jun** homodimers via ***interaction with*** **QM**/**Jif-1**."
  - "The death domain of **tumor necrosis factor receptor-1** (**TNFR1**) triggers distinct signaling pathways leading to apoptosis and **NF-kappa B** activation through **its *interaction with*** the death domain protein **TRADD**."
- "The ***interactions of*** **hsp70** ***and*** **hsp90** in intermediate **PR** complexes are shown to be distinct from their separate interactions in early **PR** complexes (**hsp70**) or in mature **PR** complexes (hsp90)."
  - "Here, a direct ***interaction between*** the activation domain of **p53** ***and*** two subunits of the **TFIID** complex, **TAFII40** and **TAFII60**, is reported."
- "Finally, we demonstrate that the anti-**CD30L** MAb M81 also completely inhibits **CD30**/**CD30L** ***interaction***."

Idea for Predicates

- A single verb may be used to express different meanings, and its meaning depends on its context (the meaning of "interaction" was the same in all of the previous sentences – PPI)
- The arguments of a single verb meaning may be expressed in various syntactic forms
- ***The parser's attachment of PPs/clauses is not reliable, and knowledge is needed about the arguments of the verb to recover the correct attachment***
- Define **predicate(s)** for each **verb meaning/syntactic variation**. A predicate (primarily) ...
  - Defines a list of verbs which may mean that predicate
  - Defines arguments (and adjuncts) labeled with semantic roles
- A semantic role defines **syntactic** and **semantic** restrictions on the **grammatical relations** which may be used to *instantiate* that semantic role – ***both must be satisfied***
- *"Our data suggest that **TR6** inhibits the **interactions of LIGHT** with **HVEM**/**TR2** and **LTbetaR** , thereby suppressing **LIGHT**-mediated HT29 cell death."*
- ***(interact-nom-of-a-with-b (verbs interact) (require theme)***

  (mod (gr (mod)) (sr thing))

  (theme (gr (imm-pp (prep of))) (sr thing))

  (cotheme (gr (pp (prep with))) (sr thing))

  (parents abstract-interact-nom))

Captures a broader meaning than PPIs since not using SRs to restrict head noun concepts of arguments.

"Pharmacological interaction of drugs with immune receptors"

- Four predicates currently exist for the "interaction" nominalization – ***The SI must choose <u>one</u> for each "interaction" clause***

The Input to the SI

- A list of clauses (including nominalizations) for a sentence, where each clause contains grammatical relations linked to phrases in the parse tree
  - The grammatical relations are candidate argument/adjuncts; at the same time they may be candidate modifiers of NPs (attach to NPs)
    - Lists of candidate subjects, antecedents of anaphors
    - Candidate PP complements, clausal complements, NP complements
- Definitions of predicates, noun relations, and noun concepts in their respective ontologies (“is-a” hierarchically structured definitions)
- A lexicon for words not in WordNet; we use both only for stemming

- **Output described more fully after example –**
  - **Clauses are linked to predicates and their arguments/adjuncts are determined**

Sentence walk-through – Parse tree

*"Our data suggest that **TR6** inhibits the **interactions of LIGHT** with **HVEM**/**TR2** and **LTbetaR** , thereby suppressing **LIGHT**-mediated HT29 cell death."*

```
(S1 (S (NP (PRP$ Our) (NNS data)) (VP (VBP suggest)
       (SBAR (IN that)

               (S (NP (NN TR6_GP_4_5)) (VP (VBZ inhibits)
                   (NP (NP (DT the) (NNS interactions))
                       (PP (IN of) (NP (NN LIGHT_GP_9_10))))
Wrong              (PP (IN with) (NP-COORD
attachment             (NN HVEM_GP_11_12_SLASH_TR2_GP_13_14)
                           (CC and) (NN LTbetaR_GP_15_16))))))
       (COMMA ,)
       (S (ADVP (RB thereby))
           (VP (VBG suppressing)
               (NP (NN LIGHT_GP_19_20_HYPHEN_mediated)
                   (NN HT29) (NN cell) (NN death)))))
(PERIOD .)))
```

Sentence walk-through – MCR post-processor – "inhibits" clause

*"Our data suggest that **TR6** inhibits the **interactions of LIGHT** with **HVEM**/**TR2** and **LTbetaR** , thereby suppressing **LIGHT**-mediated HT29 cell death."*

Two possibilities for each PP and RELATIVE:

- It attaches to the verb (argument/ adjunct)

- It attaches to a phrase (e.g., NP/ADJP)

```
(inhibits-13
 (VERB
   (MAIN-VERB inhibits inhibit)
   (VERB-TYPE VERB)
   (VOICE ACTIVE)
   (TENSE VBZ))
 (PP
   (ID 22)
   (PREP (IN with))
   (NP
     (ID 24)
     (
         (AND
           (NP (ID 25) ( (NN HVEM_GP_11_12_SLASH_TR2_GP_13_14)))
           (NP (ID 27) ( (NN LTbetaR_GP_15_16))))))))
 (OBJECTS-0
   (ID 14)
   (RELATIVE (ID 17) (CONJ ) (CLAUSE interactions-17)))
 (PP (ID 18) (PREP (IN of)) (NP (ID 21) ( (NN LIGHT_GP_9_10))))
 (SUBJECT-0 (ID 10) (NP (ID 11) ( (NN TR6_GP_4_5))))
 (PARENT suggest-6))
```

Misnomer; Candidate-clausal complement (CP); can be a relative if not CP. Since it's a nominalization, it's also a candidate object (NP complement).

Not parser attached

- The clause includes PPs and RELATIVEs that are not syntactically attached
  - Also, PPs are explicitly detached from NPs and listed with the rest of the PPs as candidate arguments/adjuncts
  - RELATIVEs are treated the same way though not explicitly detached here
  - Parser syntactic attachment is preserved via IDs

- As a result the same PP/RELATIVE appears in multiple clauses

Sentence walk-through – MCR post-processor – “interactions” clause

*“Our data suggest that **TR6** inhibits the **interactions of LIGHT** with HVEM/TR2 and LTbetaR , thereby suppressing **LIGHT**-mediated HT29 cell death.”*

Not parser attached

```
(interactions-17
  (VERB
    (MAIN-VERB interactions interact)
    (VERB-TYPE NOM)
    (VOICE ACTIVE)
    (MODIFIERS ( (ID 16) ( (DT the)))))
  (PP (ID 18) (PREP (IN of)) (NP (ID 21) ( (NN LIGHT_GP_9_10))))
  (PARENT inhibits-13)
  (PP
    (ID 22)
    (PREP (IN with))
    (NP
      (ID 24)
      (
        (AND
          (AND
            (NP (ID 25) ( (NN HVEM_GP_11_12_SLASH_TR2_GP_13_14)))
            (NP (ID 27) ( (NN LTbetaR_GP_15_16)))))))))
```

Sentence walk-through – Semantic Interpreter

*"Our data suggest that **TR6** inhibits the **interactions of LIGHT** with **HVEM**/**TR2** and **LTbetaR** , thereby suppressing **LIGHT**-mediated HT29 cell death."*

```
(inhibits-13
    (pred verb-ont inhibit asserted-clause)
    (theme
        (nom-np
            (senses (interaction (mod the) (head interactions)))
            (clause interactions-17)))
    (agent
        (candidate-list (np (senses (protein (mod ) (head TR6_GP_4_5)))))))
```

(interaction (nouns interaction)
(parents interaction-property))

IDs and Redundant nesting removed

```
(interactions-17
    (pred nom-ont interact-nom-of-a-with-b asserted-clause)
    (theme (pp (prep of)
               (np (senses (protein (mod ) (head LIGHT_GP_9_10))))))

    (cotheme
      (pp (prep with)
        (np (AND
            (np (senses (protein (mod ) (head HVEM_GP_11_12_SLASH_TR2_GP_13_14))))
            (np (senses (protein (mod ) (head LTbetaR_GP_15_16)))))))))))
```

(protein
(pattern _GP_([\d]+)_([\d]+))
(parents biological-entity))

Recovered attachment

Output of the SI

- Clauses are labeled with verb meanings (predicates) in the predicate ontology
- For the chosen predicate of each clause, arguments and adjuncts are extracted and labeled with semantic roles (using the semantic role definitions of the predicate)
  - PP attachments are determined using a combination of information from the parser and from semantic roles of predicates and subcategorization of noun concepts
  - Head nouns are labeled with concepts in the noun ontology and may be restricted by SRs
- Additional noun relations (triggered by head nouns) are outputted using a list of noun relation rules – Annotated similarly
- If a semantic role is not generated for a grammatical relation, then a grammatical role for it is still generated; If a clause is not linked to a predicate, then it will consist only of grammatical roles

Semantic Interpreter – Core of Algorithm

- Links each clause to a predicate, determines its arguments and their PP attachments
- **To choose a predicate for a clause**
  - For each predicate in the list of candidate predicates of a clause, a candidate clause is generated which has been annotated with the argument semantic roles of that predicate
    - *Determines an argument by checking the list of candidate grammatical relations for any that satisfy the argument's GRs and SRs (next slide: possible to recover from bad parser attachment)*
  - L ← the list of candidate clauses having the largest number of semantic roles
  - Delete any candidates in L not having a required semantic role; Choose an L[i] with first priority (tied candidates are discarded)
- **Attach PPs to the phrases of the semantic roles of the resulting clauses with the restriction that a PP which is already a semantic role cannot be attached**
  - Firstly, attach PPs to to NPs having a noun concept that subcategorizes for a preposition (e.g., “region” subcategorizes for “in”); possible to recover from bad parser attachment
  - Secondly, attach the rest of the PPs using the parser's attachment

Semantic Interpreter – Core of Algorithm - *Recover from attachment errors; Some GRs rely on initial SI output*

- **Recover correct attachment of a PP/RELATIVE that has been wrongly attached to an NP using the semantic roles of verb predicates**
  - If a PP/RELATIVE that is attached to the verb is a semantic role, then that PP/RELATIVE cannot be a semantic role of a verb that does not have that PP/RELATIVE attached (give preference to parser attachment)
  - Requires two passes (of previous slide): one for checking which semantic roles are syntactically attached, and one for generating the final roles
  - The second pass can generate semantic roles from PPs/RELATIVEs which are not syntactically attached

- **Some GRs rely on an initial SI output – roles with PP attachments – in order to be determined – requires two passes of the algorithm as just described**
  - Some GRs require syntactic information from ***PP attachments*** of arguments, ***which cannot be determined until semantic roles have been determined***. These GRs, in the first pass place a (reduced, non-attachment checking) restriction on the inputted grammatical relation, while in the second place, they place the full restriction on the role generated in the first pass.
    - *Above is a proposed computational efficiency improvement*
      - *In the current implementation, those clauses are actually always regenerated from scratch in the second pass, and both of the passes above happen in the second pass*
      - *The first pass for these clauses is computational overhead*

Knowledge Extractor

- **KE-1** uses semantic **extraction rules** to extract the desired relations– particular arguments/adjuncts from particular SI relations (i.e.,, argument-wise relations) ***(extraction rules = relation/argument extraction rules)***
- **KE-2** uses **filtering rules** to extract the entities (e.g., proteins) from the KE-1; ***(filtering rules = entity extraction rules)***
- **KE-3** produces n-tuples of entities (i.e., entity-wise relations) from the KE-2

KE-1 extraction rule

```
(predicate abstract-interact-nom
        ke1-abstract-interact-nom
      (theme (noun protein))
      (cotheme (noun protein)))
```

KE-2 filtering rule

```
(filter-phrase filter-event-nom
    (cond (event-nom)
          (pred -exist-in -confer -encode))
    (exclude-from-filter head-noun))
```

Extract “theme” from Predicates subsumed by abstract-interact-nom that contain a protein entitiy

event-nom – If a condition is met, the phrase is filtered

(1) extraction of interactions has taken place in nominalization
(2) nominalization has at least two semantic roles satisfied
    (“mod” is special and not counted in number of roles)

(TO-DO: conditions for event-nom should be parameterized in the rule)

Knowledge Extractor – Example 1

```
(KE-2
    (inhibits-13
        (ke1-abstract-directed-interact
        (
            (agent (np (extracted-nouns (protein TR6_GP_4_5))))
            (theme
              (filter-phrase filter-event-nom)
              (np (extracted-nouns ))))))
    (interactions-17
        (ke1-abstract-interact-nom
        (
            (theme (pp (prep of)
              (np (extracted-nouns (protein LIGHT_GP_9_10)))))
            (cotheme
              (pp (prep with)
                (np
                  (AND
                    (np (extracted-nouns
                      (protein HVEM_GP_11_12_SLASH_TR2_GP_13_14)))
                    (np (extracted-nouns
                      (protein LTbetaR_GP_15_16)))))))))))
```

```
(filter-phrase filter-event-nom
    (cond (event-nom)
          (pred -exist-in -confer -encode))
    (exclude-from-filter head-noun))
```

```
(predicate abstract-interact-nom
       ke1-abstract-interact-nom
    (theme (noun protein))
    (cotheme (noun protein)))
```

KE-2 filtering rules are executed in a depth-first traversal of the KE-1 tree. Each rule is tried on each phrase, token or relation.

*“Our data suggest that **TR6** inhibits the **interactions of LIGHT** with **HVEM**/**TR2** and **LTbetaR** , thereby suppressing **LIGHT**-mediated HT29 cell death.”*

Knowledge Extractor – Example 2 (Interaction keyword is an argument/adjunct's head noun)

Expressed in a murine myeloma , TRAP_GP_6_7 was identified as a ligand for CD40_GP_13_14 by binding to a soluble CD40_GP_19_20 construct .

KE-1 extraction rule

```
(predicate nil general-rule-interaction-property
    (theme (sr interaction-property) (noun protein))
    (cotheme (noun protein)))
```

```
(KE-1
 (identified-19
  (general-rule-interaction-property
   ( (cotheme (np (senses (protein (mod ) (head TRAP_GP_6_7)))))
    (theme
     (pp
      (prep as)
      (nom-np
        (nom-np
          (senses (ligand (mod a) (head ligand)))
          (clause ligand-25))
        (cotheme
          (pp
          (prep for)
            (np (senses
             (protein (mod ) (head CD40_GP_13_14))))))))))))
```

"nil" rule matches any predicate; it is independent of SI role names (role names are *assigned* not matched) *but subject to SRs*

Noun concepts

```
(ligand
    (nouns ligand)
    (subcat (prep for)
       (sr thing))
    (parents
     interaction-property))

(interaction-property
    (parents biological-entity
     biological-event))
```

Nom predicate

```
(ligate-nom
    (verbs ligate)
    (theme (gr (prn) (appos))
             (sr thing))
    (cotheme (gr (mod) (pp (prep for)))
             (sr thing))
    (parents
  abstract-interact-substance-nom))
```

Knowledge Extractor – Example 3 (NO PPIs – Uses filtering rules)

As expected, overexpression of either **Id3/HLH462** or **ITF-2b** effectively inhibited the activation of the muscle-specific **creatine kinase** promoter by the myogenic transcription factor **MyoD**.

```
(KE-2
 (inhibited-31
  (ke1-abstract-directed-interact
   (
    (agent
     (filter-phrase
      filter-attachments-of-np-with-
       disallowed-head))
    (theme
     (filter-phrase filter-event-nom)
     (np (extracted-nouns ))))))
 (activation-35
  (ke1-abstract-interact-nom
   (
    (theme
     (pp
      (prep of)
      (filter-phrase filter-np-with-
         disallowed-head-concept)))
    (cotheme
     (pp (prep by)
      (np (extracted-nouns (protein
MyoD_GP_25_26)))))))))
```

```
(filter-phrase filter-attachments-of-np-with-
                    disallowed-head
    (cond
        (label np np-prn prn np-appos nom-np)
        (np-child-0
            (h-concept exon mutant cotransfection
             formation isoform absence neither
             coexpression overexpression)))
    (exclude-from-filter child-0))
```

ML Task – Template filtering rules for generating features for entities

*As expected, overexpression of either **Id3/HLH462** or **ITF-2b** effectively inhibited the activation of the muscle-specific **creatine kinase** promoter by the myogenic transcription factor **MyoD**.*

```
(filter-phrase T-filter-attachments-of-np-with-
                    disallowed-head
      (cond
            (label np np-prn prn np-appos nom-np)
            (np-child-0
                  (h-concept <T>)))
      (exclude-from-filter child-0))
```

Inserted in the KE-2 tree corresponding to the KE-1 tree where rule was checked

```
(filter-phrase
  <T>
  T-filter-phrase-and-attachments-by-head-concept
  h-concept
  overexpression)
```

A depth-first traversal to each entity yields the list of condition instances which could filter that entity – if it were actually a rule

```
(filter-phrase
  <T>
  T-filter-phrase-by-mod-word
  mod-word
  muscle-specific)
```

```
(filter-phrase
 <T>
 T-filter-phrase-by-
        head-concept
 h-concept promoter)
```

- Suppose the rule's conditions are checked on "overexpression of..." NP.
- If "label" is satisfied, the np-child-0 will be checked for condition "h-concept <T>".
  - *A <T> condition is only checked if all previous conditions are satisfied*
- An "h-concept <T>" condition will not filter, but instead generates all possible condition instances which could filter the phrase – if such a rule were used in actuality
  - An "h-concept <T>" condition generates all "h-concept H" condition instances, where H is a head concept of the phrase

ML Task: Learning to filter entities

- **What is classified?**
  - Outputted entities of the system (i.e., involved in a relation)
  - Such entities are either TPs or FPs with respect to the gold-standard – we want to filter out the FPs while not filtering the TPs
- **What are the classes?**
  - Don't filter (0) – An entity is classified as 0 in the training data **if it is involved in any gold-standard interaction**
  - Filter (1) – An entity is classified as 1 in the training data **if it is not involved in any gold-standard interaction (i.e., the entity is a FP)**
- **What are the features of entities?**
  - The list of template condition instances on the depth-first traversal of the KE-2 tree to that entity
  - The list of template condition instances of the other entities – those that are involved in an outputted relation with the entity
    - The essential/desired filtering feature may only be present in the DFT of one of the entities in the PPI – e.g., “promoter” head noun
    - But if both are FPs, both are to be classified as 0 – this essential feature must be present in both entities
    - This feature sharing causes the essential/desired feature to be weighted towards class 0 in training

Experimental Setup – Template filtering rules

```
(filter-relation T1 (cond (ex-rule general-rule-interaction-property) (arg-head <T>)))

(filter-phrase T2 (cond (label np np-prn prn np-appos nom-np) (np-child-0 (h-concept <T>))))

(filter-phrase T3 (cond (label np np-prn prn np-appos nom-np) (np-child-0 (m-concept <T>))))

(filter-phrase T4 (cond (label np np-prn prn np-appos nom-np) (np-child-0 (mod-word <T>))))

(filter-phrase T5 (cond (m-concept <T>)))

(filter-phrase T6 (cond (mod-word <T>)))

(filter-phrase T7 (cond (h-concept <T>)))

(filter-phrase T8 (cond (alt-ex-head <T>)))
```

Some of the rules in the manual ruleset were replaced by template rules to create a new ruleset, the "template ruleset", which preserves the original ordering of the manual ruleset.

Templates determine which features are to be generated for outputted entities.

Experimental Setup

- **Inter-annotator disagreement – our judge and AIMed are the two annotators**
  - We only consider PPIs that our expert judge agrees with – throw out disagreements
  - If True PPI and judge disagrees, system is not penalized for missing it, and not given credit for recovering it; if False PPI and judge disagrees, system is not penalized for getting it, and not given credit for not getting it (in any case, TNs aren't used to compute precision/recall)
- Entities not outputted by the system, entities with no condition instances (extracted features) and entities in an interaction in the IAD file are all classified as "0" and have only a single special "nil" feature which no other entities have.
- The total number of features is 692 including the "nil" feature.
- There are 111 entities labeled as "filter" (1) in the training data.
- There are 1361 entities labeled as "no filter" (0) in the training data, but only 1072 of them have features other than the "nil" feature.

10-fold stratified CV on AIMed, using 9 parts for training and 1 part for testing in each of the 10 folds – *Used SVM with linear kernel from scikit-learn, default parameters*

Stratified = equal proportion of classes in each of the ten datasets as in the original dataset; the split for each of the ten datasets is roughly 10 to filter and 100 to not filter

Baseline systems

	Manual rules **(No-test IAD)**	Template ruleset – The filtering rules to be replaced by the classifier (for which there correspond templates) are not included **(No-test IAD) – Upper bound on recall for the learned classifier**
Precision	**96.7**	86.8
Recall	69.3	71.2
F-score	**80.8**	78.2

	Manual rules **(Test IAD) – For comparison with others' AIMed scores**	Template ruleset – The filtering rules to be replaced by the classifier (for which there correspond templates) are not included **(Test IAD) – Upper bound on recall for the learned classifier**
Precision	89.0	80.5
Recall	68.8	70.8
F-score	77.6 ***S.o.A F-Score = ~65***	75.3

Experiment 1 – Classification accuracy of outputted entities

	# features	**Average classification accuracy of testing data using stratified 10-fold CV**
RFECV (Train IAD)	**285**	***Too similar... But Next slide*** **0.97**
RFECV (No-Train IAD)	**285**	***Shows IAD vs. not are*** **0.97**
All features (Train IAD)	692	***not identical models*** 0.97
All features (No-train IAD)	692	0.97

No-train IAD means to use the IAD and **not train** any entities in an outputted IAD interaction (they are classified as 0 and given the “nil” feature)

Train IAD means **not** to use the IAD file and **train on all** outputted entities with features

RFECV = Recursive feature elimination with cross-validation. Initial training yields a weight for each feature with high positive or negative values indicating more Importance. Features are recursively deleted from the data, and the feature set corresponding to the highest CV classification accuracy is chosen.

Experiment 2 – Performance of system on AIMed using learned classifier as a post-processor

	RFECV (No-train IAD, No-test IAD)	**RFECV (train IAD, No-Test IAD)**	RFECV (train IAD, Test IAD)	RFECV (No-train IAD, Test IAD)
Precision	93.1 (manual = 96.7)	**93.5**	88.1	85.8
Recall	**71** (manual =69.3)	70.9	70.5	70.6
F-score	**80.6** *(Manual =80.8)*	**80.6**	78.3	77.5

	All features (No-proba-cutoff) (No-test IAD)	All features (proba-cutoff 0.92) (No-test IAD)
Precision	92.6	**94.1**
Recall	71.1	69.4
F-score	**80.5**	79.9

Manual rules in conjunction with RFECV (train IAD); (No-Test IAD)
97.9
69.3
81.1

Discussion

RFECV produces only a slight increase in F-score (0.1%) compared to All features (No-proba cutoff)

Training on the IAD or not produces similar results (but not identical, as RFECV table shows)

Ignoring IAD interactions during system evaluation produces better results in either case

The ML filter only classifiers interactions outputted by the system, i.e., TP and FP ; most of the misclassifications of the system come from FN, so even though the ML filter has 0.97 accuracy, the overall improvement in F-score compared with template ruleset baseline is only 2.4%

Manual rules plus RFECV classifier is best – does not reduce recall from manual ruleset, but increases precision by 1.2%

The intention of the filtering-classifier is to (1) reduce the workload in applying it to a new domain, (2) be able to generalize better than the current manual ruleset when tested on a different corpus

Future Work

Recent experiments on LLL show such stringent filtering rules are not needed, as they are needed for AIMed – thus the filtering rules are corpus/annotation-policy dependent

With minimal filtering rules, and NO ML-filter:

- On the initial run on LLL: 100% precision, 12% recall
- After adding some new predicates and noun concepts: 96% precision, 58.5% recall, 72.7% F-score
- LLL has a lot of modifiers that trigger interactions – an unhanded construction

Need a NER system for applying the system to real articles

- Evaluation of NER is problematic using AIMed – the NER tags must correspond *exactly* to the AIMed tag

System can easily be extended to extract different type of events besides PPI – such as up-regulation of gene expression

- The BioNLP challenge (based on GENIA corpus) requires a more fine-grained relation extraction