Shock Information Extraction system

11/23/2009

Mohammed Alshayeb
Sean Neilan
Laura Christiansen

Goal of Project:

- Using abstracts of research papers to extract findings reported in biomedical texts.
- Most 'findings' concern causation (e.g. "X caused Y", "X influenced Y").

Data Set

- We collected the abstracts submitted to the 2009 conference, pre-processed them by Metamap to extracted the named entities of certain semantic types.
- Semantic types are: 0. Drug/Chemical Compound/Therapeutic Modality

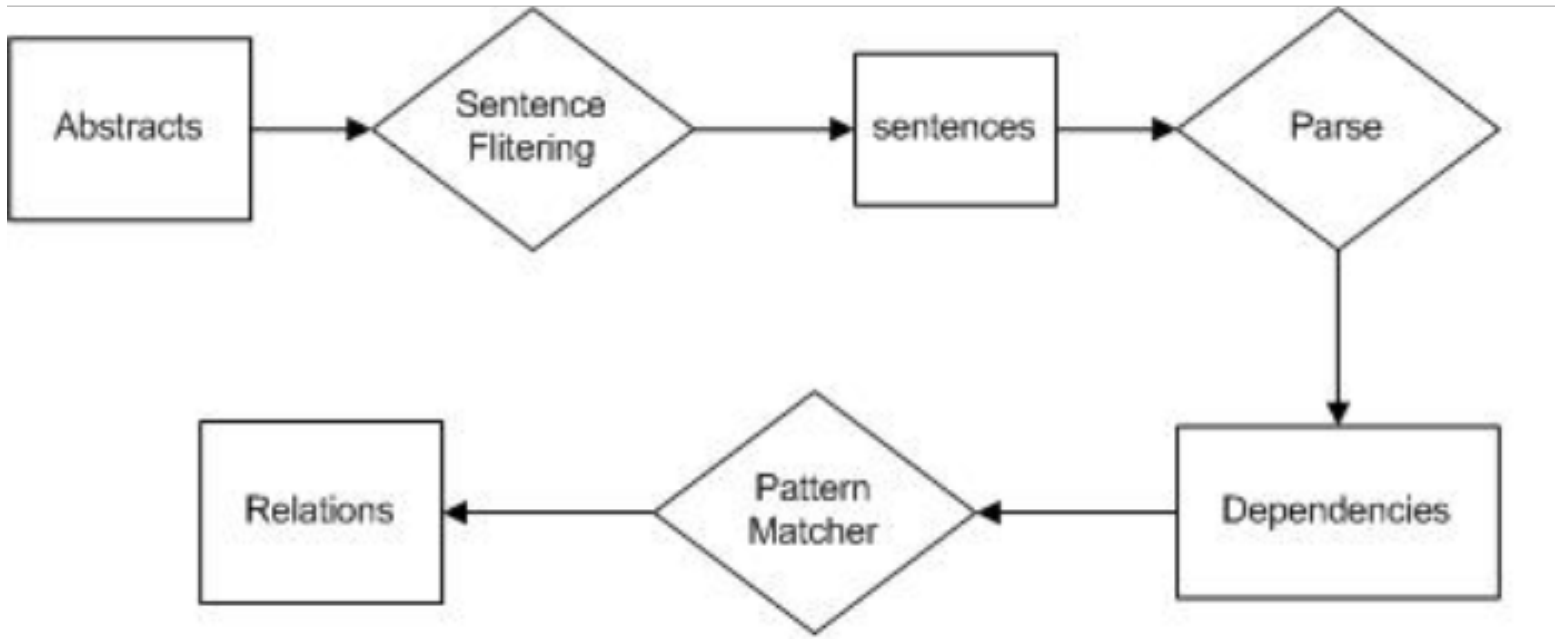
1. Molecule
2. Cell Type
3. Condition
4. Experimental Platform

Done by Dr. Gary and Emilia Apostolova.

The processing of abstracts by Shock IE System

- Step 1 (Sean): Text is fed into MetaMap. Goal: some of the words in the text should be replaced with named entity categories, plus these named entities' parts of speech.
- Step 1 Alt (Laura): Replace MetaMap with a trainable NER solution, in order to allow for the correction of NER errors.
- Step 2 (Sean & Mohammed): Output from Step 1 is fed into Stanford Dependency Parser (aka Stanford Partial Parser).
- Step 3 (Mohammed): The output of the Stanford Parser will be fed into a pattern matcher or rule-based system. This system will try to match either the sentence structure or the dependencies against patterns to produce a set of propositions expressed by the sentence.

IEShock Chart



Step 1 Alt: Initial Work

- Looked at biomedical NER tools
  - AbGene, AllAGMT, GAPSCORE, ABNER

- Modified ABNER source code for Shock Project
  - Tested MALLET v1.4 and v2.0
  - Created lexical scanner code with JLex
  - Added additional orthographic rules for NER
    - Ex: Picking out entities with suffixes -cytes, -cyte, -virus, -oid, -mRNA, -ase, -some, and -sis

Step 1 Alt: Testing

- Program can now run as a substitute first step but needs evaluation
- Assign abstracts to n stratified folds
  - Compute entity distribution for abstracts
  - Modified k-means algorithm assigns to folds

- Run n-fold cross validation
  - Use ABNER to train and test based on folds
  - Calculate and output accuracy results

Sample Prelim Testing Output

RESULTS FOR DATASET OF 10 ABSTRACTS:

Instances of DRUG_CHEMICAL_COMPOUND_THERAPEUTIC_MODALITY: 174
0.2150803461063040 8%
Instances of CELL_TYPE: 108
0.1334981458590853%
Instances of CONDITION: 382
0.4721878862793572%
Instances of GENE: 50
0.06180469715698393%
Instances of MOLECULE: 95
0.1174289245982694 7%

PRELIM DATA FOR 3 FOLDS (Stratified Cross-Validation)

Fold 1:
Abstracts: 0 3 6
Instances of DRUG_CHEMICAL_COMPOUND_THERAPEUTIC_MODALITY: 53
0.2314410480349345%
Instances of CELL_TYPE: 20
0.08733624454148471%
Instances of CONDITION: 119
0.519650655021834%
Instances of GENE: 26
0.1135371179039301 3%
Instances of MOLECULE: 11
0.04803493449781659 5%

Fold 2:
Abstracts: 8 1 7 9
Instances of DRUG_CHEMICAL_COMPOUND_THERAPEUTIC_MODALITY: 76
0.2428115015974440 8%
Instances of CELL_TYPE: 54
0.1725239616613418 5%
Instances of CONDITION: 148
0.4728434504792332%
Instances of GENE: 7
0.022364217252396165%
Instances of MOLECULE: 28
0.08945686900958466%

Fold 3:
Abstracts: 2 4 5
Instances of DRUG_CHEMICAL_COMPOUND_THERAPEUTIC_MODALITY: 45
0.1685393258426966 2%
Instances of CELL_TYPE: 34
0.1273408239700374 6%
Instances of CONDITION: 115
0.4307116104868914%
Instances of GENE: 17
0.06367041198501873%
Instances of MOLECULE: 56
0.2097378277153558 2%

Sample ABNER Tagging

Initial|O studies|B-DRUG_CHEMICAL_COMPOUND_THERAPEUTIC_MODALITY in|O C57BL|O /|O 6|O mice|O demonstrated|O that|O increasing|O or|O decreasing|O bioavailable|O IGF-I|O within|B-DRUG_CHEMICAL_COMPOUND_THERAPEUTIC_MODALITY muscle|B-CELL_TYPE by|O local|O administration|O of|O either|O Leu24|O Ala31|O IGF-I|O or|O IGF|O binding|O protein|O (|O IGFBP|O)|O -1|O ,|O respectively|O ,|O produced|O prog.|O ,|O phosphorylation|O of|O 4|O E-BP1|O and|O S6K1|O)|O of|O protein|O synthesis|O .|O

Thus|O ,|O muscle-directed|O IGF-I|O attenuates|O the|O sepsis-induced|O atrophic|O response|O apparently|O by|O increasing|O muscle|B-DRUG_CHEMICAL_COMPOUND_THERAPEUTIC_MODALITY protein|B-CONDITION synthesis|I-CONDITION and|O potentially|O decreasing|O proteolysis|O .|O

Sample Final Testing Output

CONFUSION MATRIX for FOLD 0

	DCCTM	CONDITION	CELL_TYPE	MOLECULE	GENE	OTHER
DCCTM	0	1	1	0	0	8
CONDITION	1	0	0	1	0	23
CELL_TYPE	0	1	0	0	1	3
MOLECULE	0	1	0	0	0	0
GENE	0	0	0	0	0	0
OTHER	25	84	25	33	12	830

CONFUSION MATRIX for FOLD 1

	DCCTM	CONDITION	CELL_TYPE	MOLECULE	GENE	OTHER
DCCTM	0	0	0	0	0	6
CONDITION	6	1	1	1	0	16
CELL_TYPE	0	0	0	0	0	2
MOLECULE	0	0	0	0	0	1
GENE	0	0	0	0	0	0
OTHER	65	104	26	21	19	651

AVERAGE CONFUSION MATRIX

	DCCTM	CONDITION	CELL_TYPE	MOLECULE	GENE	OTHER
DCCTM	0	1	0	0	0	8
CONDITION	2	1	0	0	0	17
CELL_TYPE	0	0	0	0	0	1
MOLECULE	0	0	0	0	0	0
GENE	0	0	0	0	0	0
OTHER	46	91	26	25	12	701

Sentence Filtering

- Find **Conclusion** word, then start filtering sentences.
- Select any sentence has two name entities.
- An interaction word that is a parent of both name Entities in a tree is saved to be used weighting and evaluation of the relations.
- TF implemented but not used.
- Replace pronoun by name entity (manually) .

Parsing

- Parse candidates sentences by Stanford Parser and get dependencies.
- Parse single sentence at a time.
- By parsing single sentence, may cause missing relation with next sentence.
- Tree or checking the beginning of next sentence if we have conjunction and next sentence has name entity may solve it.

Pattern Matcher

Relative Frequency	Category	Lexico-Syntactic pattern	Rules
=~40%	Verb	E1 Verb E2 X established Y	nsubj(verb,NE1) dobj(verb,NE2)
=~25%	Infinitive	E1 to verb E2 X plans to acquire Y	xsubj(Verb,NE1) dobj(Verb,NE2)
16%	Verb+Prep	E1 Verb Prep E2 X moved to Y	nsubj(verb,NE1) prep(verb,NE2)
1%	Coordinate(V)	E (and|,) E2 Verb X,Y merge	nn(verb,NE1) oppos(verb,NE2)

meaning of dependencies type:
http://nlp.stanford.edu/software/dependencies_manual.pdf

Pattern Matcher

- Same sentence can have more then one rule, which relation to select.
- If we know which rule have strong relation between named entities, we should apply it.
- Two Rules are used:
  - ConnectedRule: verb and matching name entities should be directly connect with a single edge.
  - NearestRule: verb and the matching name entities should be connected to each other, directly or indirectly with n edge hops, in either direction.

Algorithm

inputs: Abstracts tagged with NEs and Domains
outputs: Binary relations between NEs

foreach: abstract **do** sentence filtering:
    foreach: sentence **do** dependency parsing:
        -apply Pattern matcher by applying all rules.
        -save relations
    end
end

Output

Sentence: Conclusion: These results suggest a role for oxidative stress in the MM-CONDITION/NN/consumption of MM-MOLECULE/NN/fibrinogen during hemorrhagic shock
Dependency: nsubj(suggest-5, results-4) -- dobj(suggest-5, role-7)
Match: suggest-5(results-4, role-7)

Sentence: Conclusions: These results support the use of PPG waveform analysis as a potential diagnostic tool to detect clinically significant hypovolemia prior to the onset of cardiovascular decompensation
Dependency: nsubj(detect-18, use-7) -- dobj(detect-18, hypovolemia-21)
Match: detect-18(use-7, hypovolemia-21)

Sentence: cells improve DNA
Dependency: nsubj(improve-2, cells-1) -- dobj(improve-2, DNA-3)
Match: improve-2(cells-1, DNA-3)

Sentence: John settlement with Jennifer
Dependency: dep(John-1, settlement-2) -- prep_with(settlement-2, Jennifer-4)
Match: settlement-2(John-1, Jennifer-4)

Sentence: Antibiotic moved to Cell
Dependency: nsubj(moved-2, Antibiotic-1) -- prep_to(moved-2, Cell-4)
Match: moved-2(Antibiotic-1, Cell-4)

Sentence: Chase,USBank merge
Dependency: nn(merge-4, Chase-1) -- appos(merge-4, USBank-3)
Match: merge-4(Chase-1, USBank-3)

Sentence: tpck/MM-MOLECULE/NN or L-NAME causes hemorrhagic_shock/MM-CONDITION/NN
Dependency: nsubj(causes-4, tpck/MM-MOLECULE/NN-5) -- dobj(case-4,hemorrhagic_shock/MM-CONDITION/NN-7)
Match: case-4(tpck/MM-MOLECULE/NN-5, hemorrhagic_shock/MM-CONDITION/NN-7)

Infrastructure Progress & Plans

Manually correct Metamap output using Djangology for use in Abner

- Metamap annotations aren't correct
  - Need correct annotations to produce a working pattern matcher & make Abner a replacement for Metamap
- To fix this
  - Import Metamap annotations into Djangology
    - Djangology, a web based annotator and annotation comparison tool
  - Manually correct metamap annotations
    - Since Djangology is web based, the task can be split amongst multiple people ;)
  - Automagically export corrected annotations into Abner input format

Infrastructure Progress & Plans

  - Annotations then created by Abner can be continously corrected and used to retrain Abner with Djangology.
- Perhaps make Metamap better?
  - May not actually be worth spending time on if Djangology makes correcting annotations fast enough.
  - Tweak metamap output options? Nah.
  - Create a Hidden Markov Model to detect annotation error patterns? Definitely.
    - (Only if making patterns to detect errors after correcting annotations is worth it.)

For everything else, there's Mastercard.

Infrastructure Progress & Plans

- Metamap also incorrectly tags POS's
  - POS tags of annotated phrases can be loaded into Djangology from Metamap, quickly corrected and exported into a form readable by the Stanford Parser
  - POS tags created by Abner can also be corrected this way too.
- More patterns
  - Progress in making more patterns is limited by operation of Stanford Parser.
  - It takes something like septic_shock/MM-CONDITION/NN and turns it into 3 words: septic_shock, '/' and MM-CONDITION/NN.

Infrastructure Progress & Plans

- To fix Stanford Parser, one must programmatically mark words as different POS's rather than tacking /NN for something like noun phrase on the end.
- Putting a / between words messes it up.
- septic_shock/MM-CONDITION/NN will become MM-CONDITION and programmatically be known as a noun and be the phrase septic_shock in the pattern matcher.
- Once Stanford Parser is working correctly, will be possible to make more patterns

- Patterns to relations
  - Each pattern by itself doesn't become an outputtable relation.
  - For each pattern, one must write code that makes it a relation

The End