Open (Relation) Extraction

Some of these slides are taken from the seminar lectures available at:

http://www.dfki.de/~neumann/OIEseminar2015/index.htm

VU VRIJE UNIVERSITEIT AMSTERDAM

Open (relation) extraction

- Keep in mind! In literature, Open Relation extraction is often referred as *Open Information Extraction (Open IE)*
- **Goals**
  - extract relations from the web with no training data, no list of relations
  - Put information in semantically precise form

Motivation behind Open IE

- The Web is large and diverse => relations of interest are often unanticipated
- **Three challenges**
  - Automation: Creation of suitable training without supervision
  - Corpus heterogeneity: NER and other heavy linguistic technologies have limits
  - Efficiency: Traditional IE systems can take weeks to complete

TextRunner

- One of the first Open IE systems
  - Goal: Discover all possible relations in the text
  - Highly scalable
  - Assign probabilities to the extracted relations
- **Three** key modules
  - Self-supervised learner
  - Single-pass extractor
  - Redundancy-based assessor

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, “Open information extraction from the web,” in *IN IJCAI*, 2007, pp. 2670–2676.

TextRunner

Self-Supervised Learner

- Input: small corpus sample
- Output: classifier that labels candidate extractions as “trustworthy” or not

1. Step: automatically labels own training data as pos/neg extractions
   - Uses constraints on the dependency parsing of the sentence (chain cannot be too long, etc.)
2. Step: uses this labeled data to train a Naïve Bayes Classifier
   - Features are domain-independent and do not require parsing: POS sequence, number of tokens, etc.

TextRunner

Single-Pass Extractor

- Makes one pass over its corpus and extracts tuples for all possible relations
  - Tags each word with its most probably POS. Entities are found by identifying noun phrases (no link to KB)
  - Finds relations by examining the text between the noun phrases
    - Omits non-essential modifiers to verbs and nouns
      - Was originally developed by => was developed by
  - Sends candidate tuples to the classifier
    - Ones labeled as “trustworthy” are extracted and stored

TextRunner

Redundancy-based Assessor

- The system can extract different tuples that mean the same thing

1. First tuples are normalized by removing unessential modifiers
2. Identical tuples are merged together. The system counts the number of duplicates (e.g. <e1 r e2> was extracted 100 times)
3. Depending on the counts, it assigns a probability that <e1 r e2> is indeed a true relation

TextRunner

- Tested TextRunner on 9 million Web pages corpus
- Compared vs. KnowItAll (state-of-the-art traditional IE system)
  Selected 10 relations, manually selected good output

	Average Error rate	Correct Extractions
TextRunner	12%	11,476
KnowItAll	18%	11.631

- TextRunner extracted 60.5 million tuples from 133 million sentences

VU VRIJE UNIVERSITEIT AMSTERDAM

ReVerb [1]

Serious problem of open IE systems (like TextRunner)

- Many extractions are:
  - *Incoherent:* extraction phrases have no meaningful interpretation
  - *Uninformative:* extraction omits crucial information

[1] A. Fader, S. Soderland, and O. Etzioni, “Identifying Relations for Open Information Extraction,” in *EMNLP*, 2011, pp. 1535–1545.

ReVerb

Serious problem of open IE systems (like TextRunner and WOE)

- Many extractions are:
  - *Incoherent:* extraction phrases have no meaningful interpretation

Sentence	Incoherent Relation
The guide *contains* dead links and *omits* sites.	contains omits
The Mark 14 *was central* to the *torpedo* scandal of the fleet.	was central torpedo
They *recalled* that Nungesser *began* his career as a precinct leader.	recalled began

ReVerb

Serious problem of open IE systems (like TextRunner and WOE)

- Many extractions are:
  - *Incoherent:* extraction phrases have no meaningful interpretation
- Up to 13-30% output is incoherent

ReVerb

Serious problem of open IE systems (like TextRunner and WOE)

- Many extractions are:
  - *Incoherent:* extraction phrases have no meaningful interpretation
  - *Uninformative:* extraction omits crucial information

is	is an album by, is the author of, is a city in
has	has a population of, has a Ph.D. in, has a cameo in
made	made a deal with, made a promise to
took	took place in, took control over, took advantage of
gave	gave birth to, gave a talk at, gave new meaning to
got	got tickets to, got a deal on, got funding from

Ex. Faust made a deal with the devil.
(Faust, made, a deal)-(Faust, made a deal with, the devil)

ReVerb

Serious problem of open IE systems (like TextRunner and WOE)

- Many extractions are:
  - *Incoherent:* extraction phrases have no meaningful interpretation
  - *Uninformative:* extraction omits crucial information
- 4-7% (WOE/TextRunner) of the relations are uninformative
- The reason is that the extractor handles improperly verb-noun relation phrases (light verb constructor -- LVC)
- LVC: multi-word expression composed by a verb and a noun where the noun carries the semantic content

ReVerb

Serious problem of open IE systems (like TextRunner and WOE)

- Many extractions are:
  - *Incoherent:* extraction phrases have no meaningful interpretation
  - *Uninformative:* extraction omits crucial information

Proposed solution

- Introduce a syntactic constraint. Every relation must start with a verb and follow a given POS grammar

$$V \mid VP \mid VW^*P$$
$$V = \text{verb particle? adv?}$$
$$W = (\text{noun} \mid \text{adj} \mid \text{adv} \mid \text{pron} \mid \text{det})$$
$$P = (\text{prep} \mid \text{particle} \mid \text{inf. marker})$$

VU VRIJE UNIVERSITEIT AMSTERDAM

ReVerb

Proposed solution

- Introduce a syntactic constraint. Every relation must start with a verb and follow a given POS grammar

$$V \mid VP \mid VW^*P$$
$$V = \text{verb particle? adv?}$$
$$W = (\text{noun} \mid \text{adj} \mid \text{adv} \mid \text{pron} \mid \text{det})$$
$$P = (\text{prep} \mid \text{particle} \mid \text{inf. marker})$$

- Only a verb: *Invented*
- A verb followed by a preposition: *Located In*
- A verb followed by a noun: *Has atomic weight of*

ReVerb

Is it enough?

- **No,** sometimes this grammar captures very specific relations

[**The Obama administration**]
is offering only modest greenhouse gas reduction targets at
[**the conference**]

- To fix this, we can add a **lexical constraint:** Each relation must appear at least *n* times with different arguments

ReVerb

How much do we loose?

First test set

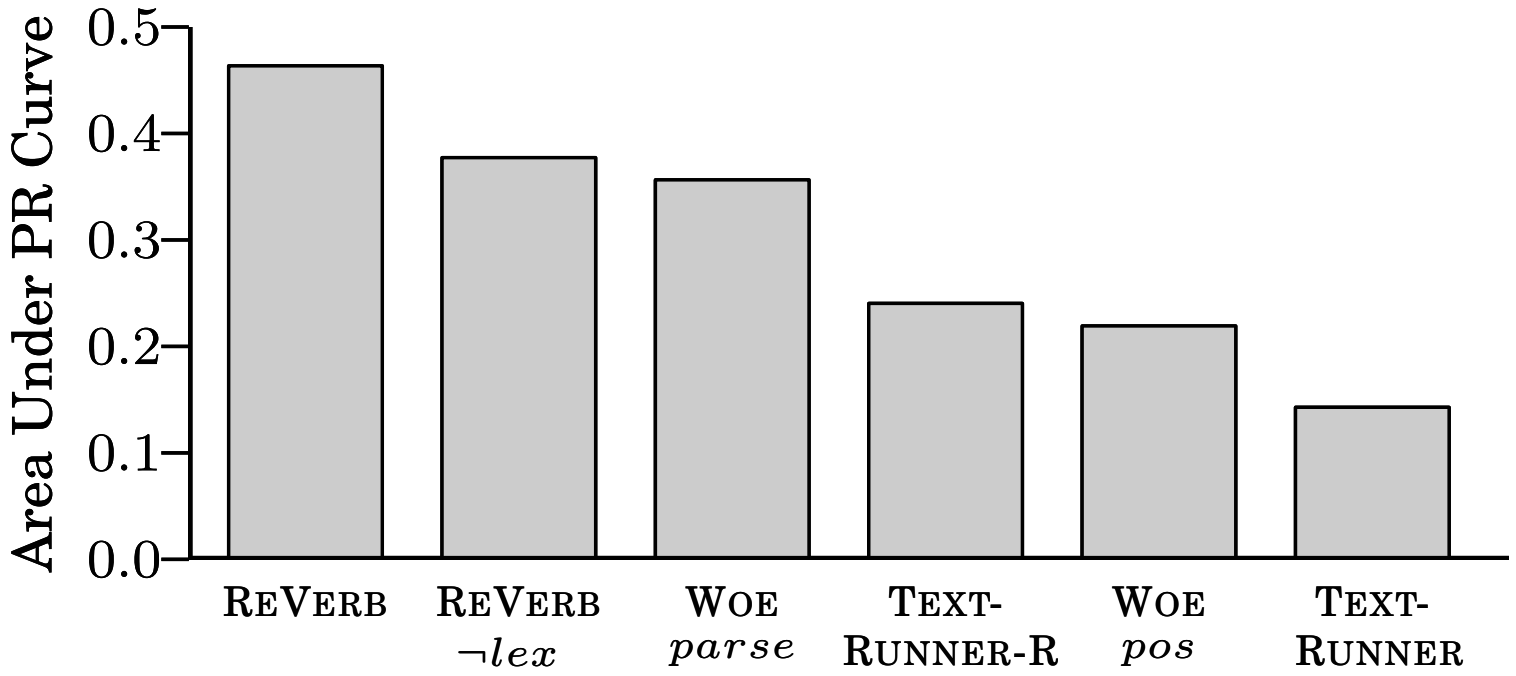
- Random web pages
- 300 sentences
- 327 verb relation phrases

	Binary Verbal Relation Phrases
85%	Satisfy Constraints
8%	Non-Contiguous Phrase Structure Coordination: X is produced and maintained by Y Multiple Args: X was founded in 1995 by Y Phrasal Verbs: X turned Y off
4%	Relation Phrase Not Between Arguments Intro. Phrases: Discovered by Y, X . . . Relative Clauses: . . . the Y that X discovered
3%	Do Not Match POS Pattern Interrupting Modifiers: X has a lot of faith in Y Infinitives: X to attack Y

VU VRIJE UNIVERSITEIT AMSTERDAM

ReVerb

Evaluation



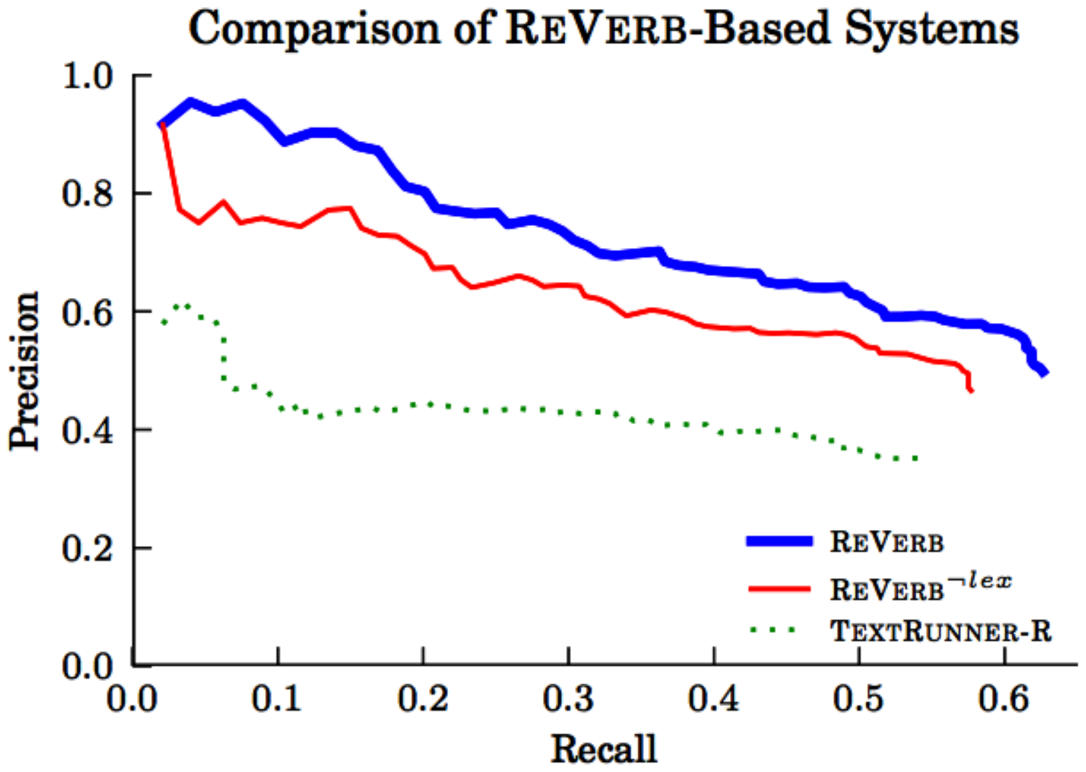
VU VRIJE UNIVERSITEIT AMSTERDAM

ReVerb

Evaluation

500 sent.
from Web

2 judges,
0.68 agr.



VU VRIJE UNIVERSITEIT AMSTERDAM

PATTY [1]

- ReVerb constraints patterns to verbs or verb phrases
- Can we learn more arbitrary relation patterns?
- **Patty:** system to compile relational patterns from a corpus, and to impose a semantically typed structure on them
- *Main idea:* WordNet groups nouns, verbs and adjectives into sets of synonyms. We can do the same with relations
  - *E.g. <X is romantically involved with Y>* is a synonym of *<X is dating Y>*
  - *<X is romantically involved with Y>* is subsumed by *<X knows Y>*

VU VRIJE UNIVERSITEIT AMSTERDAM

[1] N. Nakashole, G. Weikum, and F. Suchanek, “PATTY: a taxonomy of relational patterns with semantic types,” in *EMNLP*, 2012, pp. 1135–1145.

PATTY

System overview

- First apply the Stanford Parser to individual sentences and obtain dependency paths

"Winehouse effortlessly performed her song Rehab."

nsubj(performed-3, Winehouse-1)
advmod(performed-3, effortlessly-2)
poss(Rehab-6, her-4)
nn(Rehab-6, song-5)
dobj(performed-3, Rehab-6)

See here for list of dependency relations

PATTY

System overview

- Then it detects the named entities in the sentence

"**Winehouse** effortlessly performed
her song **Rehab (song)**."

- Finally it finds the shortest path in the dependency tree between them

Winehouse *nsubj* performed *dobj* Rehab

- Some constraints are imposed to ensure the relation is subject-relation-object

VU VRIJE UNIVERSITEIT AMSTERDAM

PATTY

- The words in the shortest path become the *textual* pattern
- **Problem:** textual patterns are tied to the particular surface form of the text
- Patty transforms textual patterns in other patters called *syntactic-ontologic-lexical* patterns (SOL)
- **Definition:** A SOL pattern is an abstraction of a textual pattern that connects two named entities

PATTY

- A SOL pattern contains:
- [pos] = a word of a certain POS *pos*
- * = a sequence of zero or more words
- <class> = an instance of the class *class*
- Every pattern contains at least two instances, which identify the entities

Example:

<person>'s [adj] voice * <song>

Matches the string: "Amy Winehouse's soft voice in "Rehab"

PATTY

Some definitions

- A pattern *P* is syntactically more general than *Q* if every string that matches *Q* matches also *P*
- A pattern *P* is semantically more general than *Q* if the set of entities that match *Q* also match *P*
- If P is semantically more general than Q and Q is semantically more general than P then P and Q are *synonymous*.

PATTY

Pattern extraction

- SOL patterns are extracted from the textual patterns:
  - First the patterns are split into *n-grams*
  - Then SOL patterns are constructed by taking the most popular n-grams and by replacing the other words by wildcards

"Winehouse effortlessly performed her song Rehab (song)."

<Winehouse, effortlessly> => not popular

<effortlessly, performed> => popular

<performed, her> => not popular

- Entities are replaced by the class they belong

<Person> effortlessly performed <song>

VU VRIJE UNIVERSITEIT AMSTERDAM

PATTY

- We can perform a *syntactic generalization* by replacing words with their POS tags

<Person> effortlessly performed <song>

1- <Person> [adv] [vrb] <song>
2- <Person> effortlessly [vrb] <song>
3- <Person> [adv] performed <song>

- Not all generalizations are good. For instance, 2) could match both “sing” and “re-arranges”. We must check the matched entities are the same

PATTY

Pattern Taxonomy

- Finally, patterns are arranged in a taxonomy

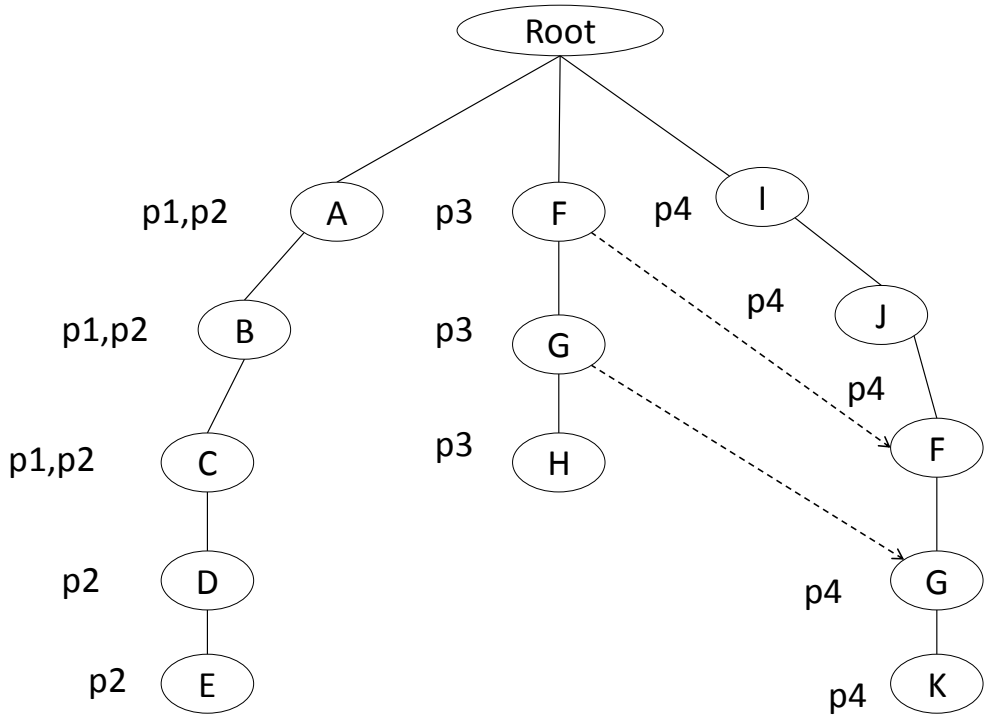
ID	Pattern Synset & Support Sets
P_1	$\langle Politician \rangle$ was governor of $\langle State \rangle$ A,80 B,75 C,70
P_2	$\langle Politician \rangle$ *politician from* $\langle State \rangle$ A,80 B,75 C,70 D,66 E,64
P_3	$\langle Person \rangle$ *daughter of* $\langle Person \rangle$ F,78 G,75 H,66
P_4	$\langle Person \rangle$ *child of* $\langle Person \rangle$ I,88 J,87 F,78 G,75 K,64

VU VRIJE UNIVERSITEIT AMSTERDAM

PATTY

Pattern Taxonomy

- The system constructs a *prefix-tree,* a well-known data structure in data mining
- Traversing the tree *bottom-up* gives us *subsumption* between patterns. E.g. P2 more generic than P1



VU VRIJE UNIVERSITEIT AMSTERDAM

PATTY

Evaluation

Corpus	Types	Patterns	Top 100	Random
NYT	YAGO2	86,982	0.89±0.06	0.72±0.09
	Freebase	809,091	0.87 ±0.06	0.71±0.09
WKP	YAGO2	350,569	0.95±0.04	**0.85**±0.07
	Freebase	1,631,531	0.93±0.05	0.80±0.08

Table 2: Precision of Relational Patterns

PATTY

Evaluation

- $\langle person \rangle$ *nominated for* $\langle award \rangle$ $\sqsupset$
  $\langle person \rangle$ *winner of* $\langle award \rangle$
- $\langle person \rangle$ *' s wife* $\langle person \rangle$ $\sqsupset$
  $\langle person \rangle$ *'s widow* $\langle person \rangle$

gold standard	PATTY	YAGO2	DBP	FB	NELL
163	**126**	31	39	69	13

Table 4: Coverage of Music Relations

Other systems

- There are other systems which we will not cover but we must mention

NELL [1]:

- Large project started at CMU by Tom Mitchell
- Goal: Extract information from the web and populate a large knowledge base.
- Learn to read better each day than the day before (LINK)

[1] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, “Toward an Architecture for Never-Ending Language Learning.,” in *AAAI*, 2010, vol. 5, p. 3.

Other systems

- There are other systems which we will not cover but we must mention

ClausIE [1]:

- Exploit linguistic knowledge about English grammar to detect clauses in the sentences
- Can extract more information than ReVerb, WOE and TextRunner

[1] L. Del Corro and R. Gemulla, “ClausIE: Clause-based Open Information Extraction,” in *Proceedings of the 22Nd International Conference on World Wide Web*, Republic and Canton of Geneva, Switzerland, 2013, pp. 355–366.

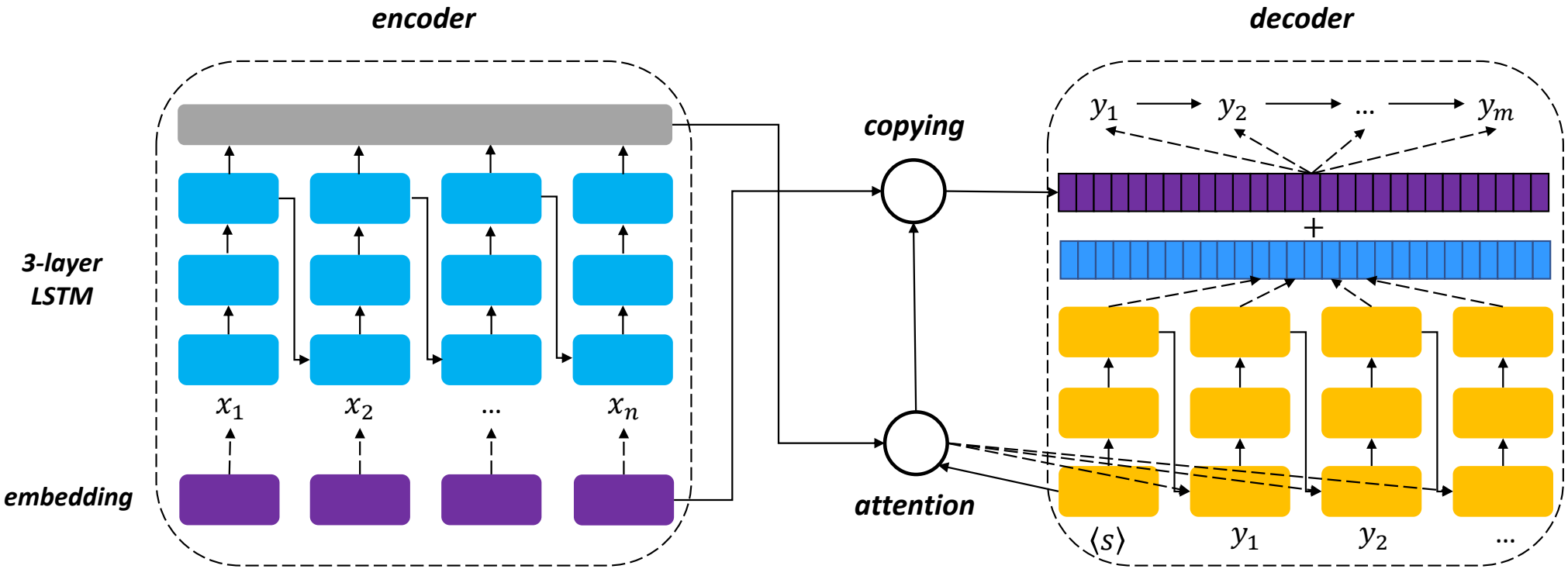
Neural Open Relation Extraction

- Recently, sequence to sequence models have been shown to be effective for open IE [1]
- The idea is to train a neural network that “translates” sequences into the corresponding extractions
- For instance: “*deep learning is a subfield of machine learning*” => “⟨arg1⟩ *deep learning* ⟨/arg1⟩ ⟨rel⟩ *is a subfield of* ⟨/rel⟩ ⟨arg2⟩ *machine learning* ⟨/arg2⟩”

[1] L. Cui, F. Wei, and M. Zhou, “Neural Open Information Extraction,” *arXiv:1805.04270 [cs]*, May 2018.

Neural Open Relation Extraction

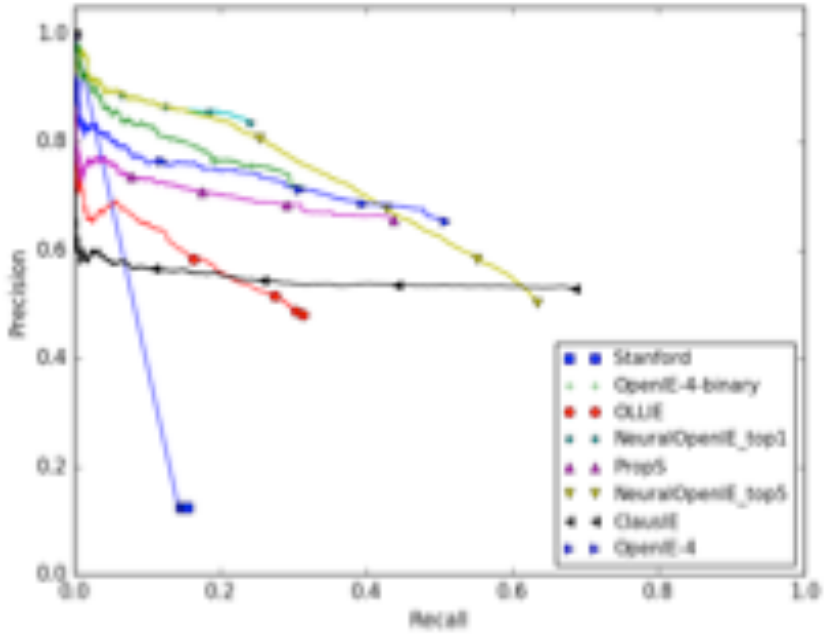
- Reuse *encoder-decoder* neural architectures which are the state-of-the-art for automatic text translation

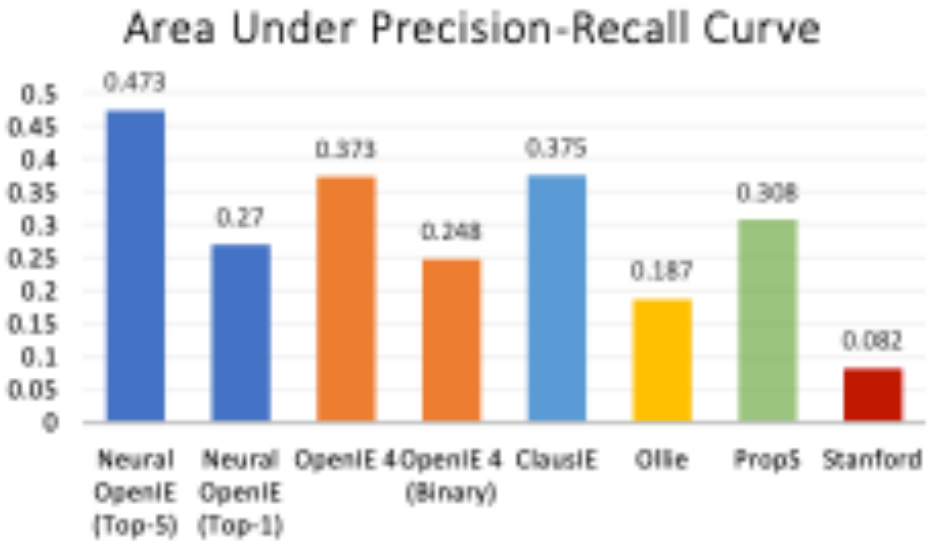


VU VRIJE UNIVERSITEIT AMSTERDAM

Neural Open Relation Extraction

- Network is trained using the output of a conventional extractor (Open IE4)





VU VRIJE UNIVERSITEIT AMSTERDAM