Metaphor Interpretation Using Paraphrases Extracted from the Web

Bollegala & Shutova 2013

Ozan Yilmaz

26th June 2019

ICL Ruprecht-Karls-Universität Heidelberg

Aufbau

1. Einleitung
2. Methoden
3. Experimente
4. Ergebnisse & Diskussion
5. Fazit

Einleitung

Einleitung

- Metaphern gängiges linguistisches Konzept -> Jeder dritte Satz in normalen Texten
- Anwendungen wie MT (zB. wörtliche Übersetzungen), Opinion Mining, IE und recognizing textual entailment könnten von Metapherverarbeitung profitieren
- Metaphern benutzt bei starken Meinungen (IE/ Opinion Mining)

Definition

- Metapher = Konzept von Domäne A in Domäne B benutzen
  - How can I **kill** a process?
  - *Computational process* als lebendig angesehen
  - *kill* stellvertretend für *terminate*
  - *Computational process* als **Target-Konzept** und *lebendiges Wesen* als **Source-Konzept**

$\Rightarrow$ Mapping zwischen Domänen ermöglicht Benutzung von Metaphern

Metapher oder Wörtlich?

- Wort wird als Metapher annotiert, falls eine grundlegendere Bedeutung des Verbs im Kontext möglich
- Eine Bedeutung grundliegender (more basic), wenn:
  - konkreter
  - Beziehung zu körperlicher Aktion
  - präziser
  - historische Reihenfolge

Definition der Metapherinterpretation

- Gegeben: Verb M metaphorisch mit Nomen A
- Gesucht: Wörtliche Paraphrase L zum Ersetzen von M mit selber Bedeutung in Kontext mit A
- Beispiel: How can I kill a process ?
  $\Rightarrow$ How can I terminate a process ?

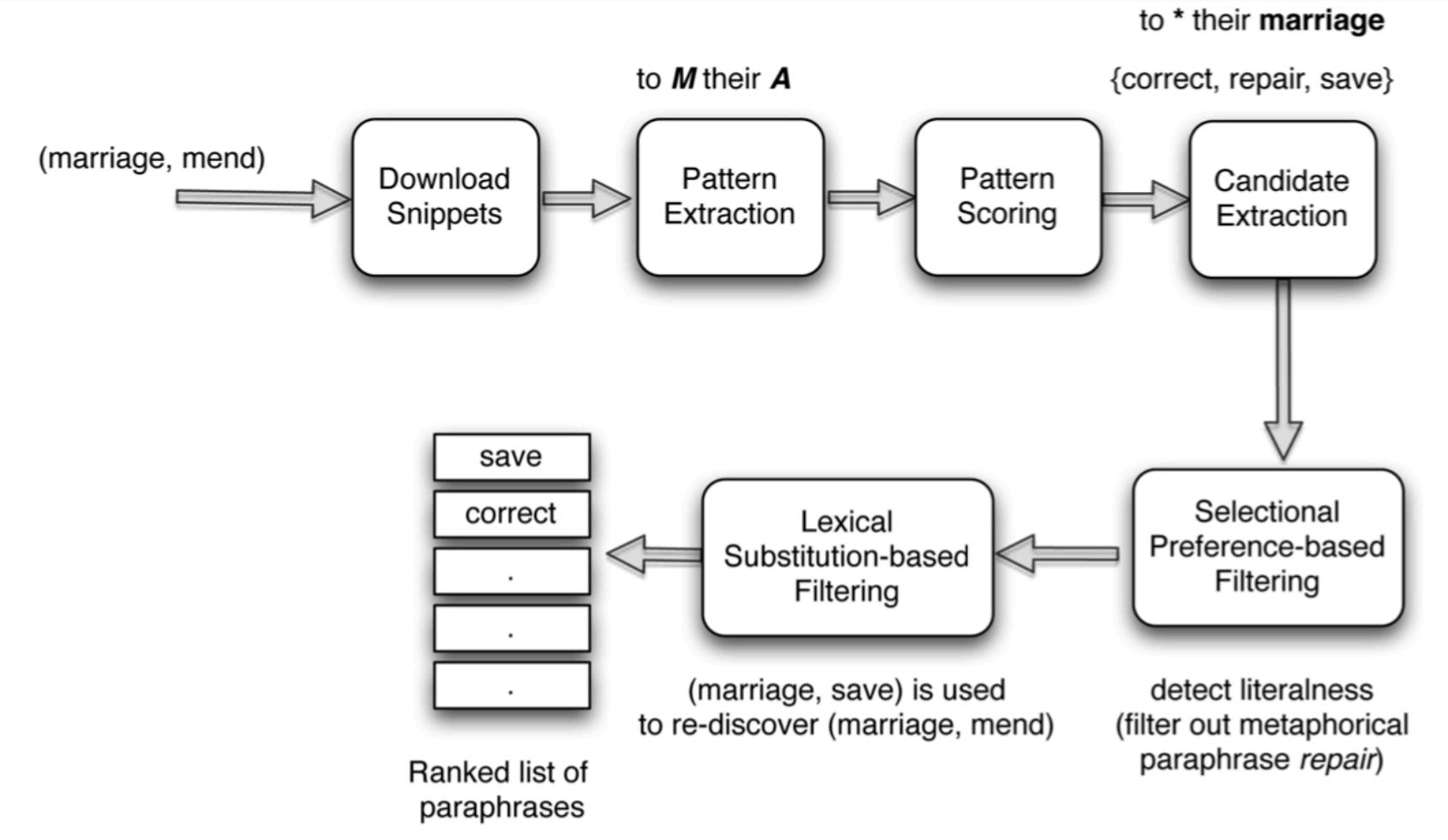
Probleme der Metapherinterpretation

- Extrahierte Paraphrasen müssen in Kontext passen (Bsp. assassinate nicht möglich für kill mit process)
- Extrahierte Paraphrasen müssen wörtliche Bedeutung tragen - nicht auch metaphorisch

Ansatz der Autoren

- Berücksichtigt die genannten Problemfaktore
- Benutzt Web Search Engine um Paraphrasen zu generiern, keine manuelle Ressource (zb. WordNet)
  - Mehr Kandidaten werden extrahiert
  - Aktuellere und kreativere Phrasen können extrahiert werden

Graph der Methode



Lexical Patterns

Beispiel ”Lexical Patterns” von Beispielrelationen mit ’bird’ und ’ostrich’:

- X = ostrich , Y = bird
- X is a large Y, Ys such as X, a large Y such as X

$\Rightarrow$ Diese werden bewertet und geranked um representativste ’lexical patterns’ für semantische Relation zu bekommen

Verwandte Arbeiten

Erste Arbeiten Metaphererkennung/-verarbeitung

- benutzten manuell angereicherte Wissensquellen(Fass D (1991), Martin JH (1990), Narayanan S (1997) ,Barnden J, Lee M (2002))
- Probleme:
  - begrenzte Abdeckung aller Fälle
  - teuer und aufwendig zu erstellen/erweitern
- Spätere Ansätze benutzten Korpora und lexikalische Ressourcen (Shutova E (2010), Veale T, Hao Y (2008)

Veale T, Hao Y (2008) Knowledge Points

Talking Points:

- Gruppe aus Eigenschaften die zu Source/Target Domain gehören
- Verwandte Informationen mithilfe von WordNet und Web
- Organisiert in Framework Slipnet:
    - Einfügen/Löschen/Ersetzen von Definitionen von Eigenschaften
      $\Rightarrow$ Dadurch Verbindung zwischen Source und Target Konzept herstellen

Beispiel Veale T, Hao Y (2008) Talking Points

W

$$\mathbf{Make-up} =>$$
$$\equiv \text{typically worn by women}$$
$$\approx \text{expected to be worn by women}$$
$$\approx \text{must be worn by women}$$
$$\approx \text{must be worn by Muslim women}$$
$$\mathbf{Burqa} <=$$

$\Rightarrow$ nicht auf Real-World Texten getestet

Shutova (2010)

Definiert Metapherintepretation als Paraphrasing-Task:

- Leitet wörtliche Paraphrasen von metaphorischen Ausdrücken im British National Corpus (BNC) ab
  - Extrahiert Gruppe von potentiellen Ersatzausdrücken in syntaktischen Konstellationen mit metaphorischem Verb V in BNC
  - Filtert Kandidaten durch Hyperonymanalyse mit WordNet
    $\Rightarrow$ wählt Verben die gemeinsamen Hyperonym haben wie V
    $\Rightarrow$ unterscheidet mit automatischen Methoden zwischen wörtlich und metaphorisch
- Überwachter Ansatz (supervised) mit WordNet -> 0.81 Accuracy

Shutova (2012)

Unsupervised Ansatz:

- Wählen der Kandidaten mit Vektorraummodell
- Selectional Preference Modell um ”Wörtlichkeit” des Ausdrucks zu identifizieren
- Evaluation auf Datensatz von Shutova 2010
  -> top-rank Precision = 0.52

  $\Rightarrow$ Sparse Data Problem -> neuer Webansatz um Problem zu lösen

Shutova und Sun (2013)

Unsupervised Learning Metapher Identifikation:

- Graphbasiertes hierarchisches Clustering von Nomen

- Precision = 0.65
  $\Rightarrow$ Nur Identifikation, keine Interpretation der Metaphern

Paraphrasenextraktion aus dem Web

In vielen Bereichen schon angewandt:

- QA, Textual Entailment Recognition, Concept Classification

- Bootstrapping mit Gruppe von Paraphrasen als lexikosyntaktische 'Pattern'
  $\Rightarrow$ Aufgabe in diesem Papier tiefgehender

Paraphrasenextraktion aus dem Web

- Wollen nur wörtliche Paraphasen von Metapherausdrücken erhalten
- Paraphrasen für ein metaphorisches Verb in vorgegebenem Kontext

  $\Rightarrow$ Trotzdem in zukünftigen Aufgaben Ansätze übernehmbar mit Filterung

Relational Similarity

Metapherinterpretation = Relational similarity zwischen Worpaaren maximieren

- Gegeben: metaphorisches Verb M, Argument A
- Gesucht: wörtliches Verb L, sodass Relational Similarity zwischen (M,A) und (L,A) so hoch wie möglich

Turney (2012)

Dual Space Modell zur Erfassung der Relational Similarity:

- Gegeben: (a,b) und (c,d)
- Domain Similarity erfassen: Lexical Patterns mit Nomen vergleichen
- Functional Similarity erfassen: Lexical Patterns mit Verben vergleichen
- Relational Similarity: Geometrisches Mittel von Domain und Functional Similarity

SemEval 2012 Task 2

Wortpaarähnlichkeiten erfassen:

- Datenset mit Zugehörigkeitsgraden zu semantischen Relationen (79) annotiert
- Nur ein System Baseline mit PMI geschlagen

$\Rightarrow$ Ansatz bisher nicht bei Metapherinterpretationen benutzt

Methoden

Methode

- Extrahiere Lexical Patterns für die semantische Relation M A
- Benutze extrahiertes Set um passende Paraphrasen zu finden (auch passend zu A)
- Wörtliche Paraphrasen mit ’Selectional Model’ auswählen
- ’Lexical Substitutabilty Test’ um Rauschen, Ambiguitäten und Antonyme rauszufiltern

Lexical Pattern Extraction

(1) Commentators claimed that she and Prince Charles had succeeded in *mending* their *marriage*.

(2) After many hours doctors finally succeeded in *saving* their *patient*.

- Lexical Pattern 'succeeded in M their A' in beiden vorhanden ⇒ mapping zwischen Source und Target Konzept(marriage vs. patient)
- Idee: Pattern finden für metaphorisches Wort und Argument -> Mithilfe von Pattern Paraphrasen finden

Lexical Pattern Extraction

Gegeben: Metaphorisches Verb M und Argument A

- Suchmaschinenanfrage ”M * * * A” -> * matcht 1 oder kein Wort
- Ziel: Finden von Webseiten die semantische Beziehung M-A beschreiben
- Double Quotes ” stellen Reihenfolge sicher
- Download Top Suchergebnisse und wählt Sätze mit A & B aus
- Wiederholen Prozess mit allen Flektionen des Verbs für Datenmenge
- Reihenfolge von M und A auch unter anderem vertauscht

Lexical Pattern Extraction

Beispielanfragen

- ”mend * * * marriage”
- ”mending * * * marriage”
- ”mended * * * marriage”

Lexical Pattern Extraction

Tools und Verarbeitung:

- Websearch mit Google REST API
- NLTK -> lowercasing, Tokenisierung und Lemmatisierung
- Ersetze Verb *M* und Argument *A* mit Placeholdern *M* und *A*
- Extrahieren n-grams (n = 3-5) mit nur jeweils einem Vorkommen *A* und *M* als Lexical Patterns

Lexical Pattern Extraction

Table 1. Extracting lexical patterns for the verb *mend* and its object *marriage*.

Query	***"mending * * * marriage"***
Sentence	Commentators claimed that she and Prince Charles had succeeded in *mending* their *marriage*
Lemmas	commentator claim that she and prince charles had succeed in ***M*** their ***A***.
Patterns	succeed in ***M*** their ***A***, in ***M*** their ***A***, ***M*** their ***A***

doi:10.1371/journal.pone.0074304.t001

Pattern Scoring

- Nicht alle extrahierten Pattern sinnvoll
- Grobe Lexical Patterns resultieren oft in inkorrekten Extraktionen -> semantic drift (Veränderung der Wortbedeutung/des Gebrauchs über Zeit)
- Viele Pattern -> Viele Webanfrange -> Langer Prozess

$\Rightarrow$ Ziel: Pattern Scoring um kleines repräsentatives Subset zu erhalten

Pattern Scoring

Annahme: word w und Lexical Pattern P extrahiert für Wortpaar (A,B)

Gesucht: Ähnlichkeit von w zur semantischen Relation (A,B), genannt Relatedness Score ->

$$\tau(w, (A, B))$$

Pattern Scoring

$$\tau(w, (A, B)) = I(w, (A, B)) - max(I(w, A), I(w, B))$$

$\Rightarrow$ Höherer Score für Wörter w die öfter mit (A,B) auftauchen statt nur mit A oder B

Pattern Scoring

$$\tau(w, (A, B)) = \text{I(w,(A,B))} - max(\text{ I(w,A) }, \text{ I(w,B) })$$

- Pointwise Mutual Information (PMI) = gibt an, ob Relation öfter abhängig vorkommt als unabhängig

Pattern Scoring

$$\tau(w, (A, B)) = I(w, (A, B)) - max(\text{I(w,A)}, I(w, B))$$

- Pointwise Mutual Information (PMI) = gibt an, ob Relation öfter abhängig vorkommt als unabhängig

Beispiel Berechnung I(w,A):

$$I(w, A) = log(p(w|A)) - log(p(w))$$

Pattern Scoring

$$\tau(w, (A, B)) = I(w, (A, B)) - max(\text{I(w,A)}, I(w, B))$$

- Pointwise Mutual Information (PMI) = gibt an, ob Relation öfter abhängig vorkommt als unabhängig

Beispiel Berechnung I(w,A):

$$I(w, A) = \text{log(p(w|A))} \text{ -log(p(w))}$$

Pattern Scoring

$$\tau(w, (A, B)) = log(p(w|(A, B))) - max(log(p(w|A)), log(p(w|B)))$$

Pattern Scoring

$$\tau(w, (A, B)) = log(p(w|(A, B))) - max(log(p(w|A)), log(p(w|B)))$$

- Approximierung der Wahrscheinlichkeiten p am Beispiel p(w|A)

$$p(w|A) \approx \frac{\text{Anzahl w in für A extrahierten Kontexten}}{\text{Für A extrahierte Kontexte}}$$

Pattern Scoring

PatScore = Summe der Relatedness Scores aller Wörter in P

$$PatScore(P) = \sum_{w \epsilon P} \tau(w, (A, B))$$

Pattern Scoring

- Nicht normalisiert mit Länge von P, da keine Verbesserung
- Nur 3 Queries nötig (A,B,(A,B)), nicht für w -> effizienter
- Pattern Scoring nicht abhängig von Webhits -> unzuverlässiges Maß

Candidate Phrase Extraction

- Unsaubere Extraktion aus Web können irrelevante Kandidaten matchen
- Einzelnes Pattern kann meist nicht alle Relationen zwischen M und A abdecken

$\Rightarrow$ CandScore für relevante Paraphrasen

CandScore

$$CandScore(c) = \sum_{P \epsilon \psi} (\text{Ext(P,c)}\ xPatScore(P)$$

- Ext(P,c) = wie oft Paraphrase c mit Pattern P extrahiert wurde
- ψ = Set von Lexical Patterns für Paar (M,A)
- Extrahierte Kandidaten werden absteigend gerankt -> Top T_c Kandidaten werden weiter bearbeitet
- Ziel: Oft extrahierte Kandidaten mit hohem Pattern Score werden hoch gerankt

Selectional Preference-based Filtering

Selectional preference Modell um wörtliche und metaphorische Kandidaten zu unterscheiden

- Bsp. M = ’accelerate change’
- Bsp. System extrahiert ’catalyse change’ und ’facilitate change’
- ’catalyse’ hätte als Source Domain eher ’CHEMICAL REACTION’ statt ’CHANGE’(Target Domain)
- ’facilitate’ hätte PROCESS(beinhaltet CHANGE) als Domain -> passt wörtlicher zu Verb

Selectional Preference-based Filtering

- Selectional Preference (SP) Verteilung der zu ersetzenden Kandidaten (S-V und V-O Relationen) mit RASP(Robust Accurate Statistical Parsing) Parser aus BNC Korpus

- SP Klassen durch Clustern der 2000 Häufigsten Nomen in 200 Cluster mit Algorithmus von Sun & Korhonen(2009)

Selectional Preference Strength (SPS)

$$S_R(v) = D(P(C|v)||P(C)) = \sum_{c \epsilon C} P(c|v) log \frac{P(c|v)}{P(c)}$$

- Kullback-Leibler Divergenz -> Unterschied zwischen zwei Wahrscheinlichkeitsverteilungen

$$D(P||Q) = \sum_{x} P(x) log \frac{P(x)}{Q(x)}$$

Selectional Preference Strength (SPS)

$$S_R(v) = D(P(C|v)||P(C)) = \sum_{c \epsilon C} P(c|v) log \frac{P(c|v)}{P(c)}$$

- P(C) = Verteilung der erwarteten semantischen Klassen -> Wie wahrscheinlich Argument zu Klasse c gehört
- P(C|v) = Verteilung der erwarteten semantischen Klassen für Verb v -> Wie wahrscheinlich Argument von v in semantischer Klasse c

$\Rightarrow$ Desto höher Unterschied zw. Verteilungen, desto mehr Information gibt Verb über mögliche Argumente an

$\Rightarrow$ Beispiel: eat sagt viel über direkte Objekte aus (normalerweise essbar), be nicht

Selectional Association

$$A_R(v, C) = \frac{1}{S_R(v)} P(C|v) log \frac{P(C|v)}{P(C)}$$

$\Rightarrow$ Wie gut passt Argument Class C zu Verb v -> Je höher, desto besser

$\Rightarrow$ Annahme: Gibt an, wie wörtlich eine Paraphrase ist

$\Rightarrow$ Top T_s Paraphrasen werden ausgewählt und weiterbenutzt

Jurafsky Speech and Language Processing Beispiel

Verb	Direct Object Semantic Class	Assoc	Direct Object Semantic Class	Assoc
read	WRITING	6.80	ACTIVITY	-.20
write	WRITING	7.26	COMMERCE	0
see	ENTITY	5.79	METHOD	-0.01

Lexical Suitability

- Bisher auf 'distributional hypothesis' von Firth JR (1957) & Harris Z (1954) verlassen:

  ⇒ "Wahrscheinlichkeit, dass M und M' Paraphrasen sind, steigt mit Vorkommenshäufigkeit von M/M' und A in gängigen 'Lexical Patterns'"

  ⇒ Problem: Antonyme werden häufig mitextrahiert -> fallen durch bisherige Filter durch

Lexical Suitability

- Gängige Lösung: Mit Parallel Korpora Antonyme rausfiltern, da nicht in allen Sprachen auf selben Target abbilden
- In diesem System -> multilinguale Ressourcen nicht angenommen bzw. benutzt
- Stattdessen: ’Lexical Suitabilty Test’

Lexical Suitability

- Antonyme folgen nicht der ’substitutability hypothesis’ (Mohammed S. et al. 2008)
- Idee: Paraphrasen für M’ und A suchen und schauen, ob M aufgefunden wird
- Falls ja -> höchstwahrscheinlich synonym im Kontext mit A

Lexical Suitability

- Alle Schritte bis zu CandScore werden mit (M',A) durchgeführt
- Falls M nicht in potentiellen Paraphrasen -> entferne M'
- Sonst werden M's nach den CandScore ranks von M in ihrem jeweiligen Durchlauf geordnet

Experimente

Datenset 1

- Annotiertes BNC Datenset von Shutova 2010
- 62 Subjekt-Verb und Verb-Objekt Konstruktionen -> Verb metaphorisch
- Beispiele Verb-Objekt: reflect enthusiasm, accelerate change, throw remark etc.
- Beispiele Subjekt-Verb: example illustrates, ideology embraces etc.
- 10 Phrases als Devset, Rest Testset
- Direkt vergleichbar mit vorherigen Shutova Experimenten

Datenset 2

- Größeres, automatisch erzeugtes Datenset
  $\Rightarrow$ Mit Shutova et al.(2013) Metapheridentifikationssystem erzeugt
  $\Rightarrow$ Fängt mit Seed Metaphern an und lernt Pattern durch Co-Clustering von Verben und Nomen

- Mit vortrainiertem Set aus BNC extrahiert

- Manuell nachbearbeitet -> 275 metaphorische Ausdrücke

- Paraphrasen extrahiert und manuell gelabelt als metaphorisch oder wörtlich

- Stellt ”Real-World Szenario” nach

Baseline und Systeme

- Baseline: Nur top 10 Paraphrasen mit CandScore
- SP: Alle Schritte bis selectional preference, OHNE lexical substitutability Test
- SP-LexSub: Alle Schritte einschließlich lexical substitutability
  $\Rightarrow$ Parameter T für einzelne Schritte: T_c=20, T_s=10 und T_l=10

Evaluation Setting 1

- 2 Unabhängige Evaluatoren mit linguistischem Hintergrund
- Bekommen Metapherausdruck und Paraphrase auf Rang 1 von allen 3 Systemen (randomisiert)
- Markieren Paraphrasen als korrekt, falls wörtlich und synonym
- System wird dann an P(1) gemessen (Anteil korrekter Paraphrasen auf Rang 1)
- Nur richtig, wenn beide Annotatoren übereinstimmen -> κ=0.66

Evaluation Setting 2

- Kontrolle des Systems mit menschlich annotiertem Goldstandard aus Shutova 2010
- 5 Annotatoren schrieben alle wörtlichen Paraphrasen auf die einfielen
- Beispiel: *brushed* aside accusations -> rejected, ignored, dismissed etc.
- Nicht alles abgedeckt $\Rightarrow$ Bestraft System evtl. unnötig

Evaluation Setting 2

Mean Reciprocal Rank (MRR):

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r_j}$$

- N = Anzahl metaphorischer Ausdrücke
- r_j = Rang erster korrekter, wörtlicher Paraphrase unter Top 5 (nach Annotoren)

Ergebnisse/Diskussion

Ergebnisse

- Evaluation von Verb-DirectObject VerbSubject und zusammen
- SP-LexSub in allen Tests statistisch signifikant besser (paired t-test mit $p<0.05$)

Ergebnisse

Table 2. Precision at rank 1 for different methods measured against human judgements.

Relation	Baseline	SP	SP-LexSub
Verb-DirectObject	0.33	0.28	0.44
Verb-Subject	0.14	0.14	0.29
Across dataset	0.30	0.26	0.42

doi:10.1371/journal.pone.0074304.t002

Ergebnisse

Table 3. Comparison of different methods against the gold standard using MRR.

Relation	Baseline	SP	SP-LexSub
Verb-DirectObject	0.122	0.217	0.265
Verb-Subject	0.088	0.166	0.219
Across dataset	0.115	0.206	0.256

doi:10.1371/journal.pone.0074304.t003

Ergebnisse

Table 4. Comparison of the different methods on the automatically collected metaphorical expressions using MRR.

Method	MRR
Baseline	0.436
SP	0.488
SP-LexSub	**0.526**

doi:10.1371/journal.pone.0074304.t004

Ergebnisse

Table 5. Top 5 paraphrases ranked for the word pair (*impose*, *decision*) with their scores.

Baseline	SP	SP-LexSub
waive (621.05)	uphold (0.31)	*enforce* (98.2)
lift (525.15)	revoke (0.21)	delay (94.5)
ease (505.14)	*enforce* (0.13)	implement (65.4)
apply (416)	implement (0.11)	uphold (65.4)
award (343.74)	postpone (0.09)	reinforce (58.2)

doi:10.1371/journal.pone.0074304.t005

Fehleranalyse

- SP schlechter als Baseline in Setting 1 aber besser in MRR
  $\Rightarrow$ Setting 1 schaut nur auf Rang 1, ignoriert Rest
- SP-LexSub auch auf großem Set besser -> robust
- Fehler bei SP oft Antonyme -> SP-LexSub Sinn erfüllt
- Verb-DirectObject immer besser
  $\Rightarrow$ nur 11 Ausdrücke für Verb-Subject, 41 für Verb-DirectObject

Fehleranalyse

Fehlerverteilung:

- Metaphorische Paraphrasen
- inpräzise Paraphrasen
- Antonyme (immer noch)
- Komplett irrelevante Topphrasen selten(13%)

Diskussion

- Schlechter als Supervised Shutova 2010
  $\Rightarrow$ P(1)=0.81, MRR=(0.63)
- SP System von Shutova 2010 nicht für unsupervised Anwendungen geeignet -> Antonyme
- Erfolgreich Problem mit LexSub behandelt und Leistung des Systems signifikant erhöht

Fazit

Fazit

- SP-LexSub relativ gute Precision (0.42) für unsupervised System
- Websuche findet viele potentielle Paraphrasen
- Aussicht:
  $\Rightarrow$ Erweitern des Systems für mehr Abdeckung
  $\Rightarrow$ Erstellen von großem Goldstandard Korpus für Metaphern mit Crowd Sourcing

Ausblick Bizzoni & Lappin 2018

Ausblick Bizzoni & Lappin 2018

- Versuchen Paraphrasenranking von menschlichen Annotatoren mit DNNs zu erreichen

- Datenset: 200 Sets mit 5 Sätzen
  $\Rightarrow$ 1 Satz Referenz mit Metapher, andere Paraphrasen
  $\Rightarrow$ Punkte 1-4 je nach Nähe zu Referenz
  $\Rightarrow$ Annotiert von 1 Autor und Pearson correlation von 0.9 mit AMT (20 Leute)

Ausblick Bizzoni & Lappin 2018

Input = 2 Sätze als Word2Vec - 1 Metapher und 1 Paraphrase
System:

- 2 Parallel CNNs und LSTMS als Encoder für Sätze
- Unified Layer merged Outputvektoren im Anschluss
- Letzter Layer mit Sigmoid Funktion für die Bewertung der Ähnlichkeit

Ausblick Bizzoni & Lappin 2018

Task 1: Binäre Klassifikation ->

- Bilde Pärchen mit Referenzsatz A und jeder Paraphrase B bis E
- Gradient Labels 1-4 -> > 2 = Paraphrase, < 2 = keine Paraphrase
- Baseline = Cosine Similarity

Ausblick Bizzoni & Lappin 2018

Model	Accuracy	F1
Baseline (cosine similarity)	50.8	10.1
Our model	**75.2**	74.6
Encoders without LSTM	64.4	64.9
Encoders without ACNN	62.6	61.5
Using CNN instead of ACNN	61.0	61.6
ACNN with 10 filters	73.4	71.7
LSTM with 10 filters	72.3	70.6
Merging via multiplication	53.4	69.6
Aligner	49.4	61.6
Aligner + our model	73.4	75.

Ausblick Bizzoni & Lappin 2018

Task 2: Paraphrasenranking ->

- Benutzen Binär trainiertes System
- Testset wird mit Sigmoidwerten geranked für 4er Sets

Ausblick Bizzoni & Lappin 2018

Measure	12-fold value	Baseline
Accuracy	67	51
Pearson correlation	0.553	0.151
Spearman correlation	0.545	0.113

Ausblick Bizzoni & Lappin 2018

Ergebnis:

- Tatsächlich kann Ranking erstellt werden mit binär trainiertem Algorithmus
- Nicht selbstverständlich, da hätte überpolarisieren können
  $\Rightarrow$ Erfolgreich double transfer learning angewandt
- Runterbrechen der Vektordimensionalität erlaubt Abstraktion zu wichtiger Semantik
  $\Rightarrow$ Evtl. auch Grund warum nicht überpolarisiert wird?

Fragen?

Vielen Dank die Aufmerksamkeit!