Regelbasierte Koreferenz mit BART

Algorithmen und Implementation zum Softwareprojekt im SS14

Julian Baumann, Xenia Kühling, Sebastian Ruder

11. August 2014

1 BART im Vergleich zu Stanford

BART unternimmt automatische Koreferenzresolution mithilfe einer modularen Pipeline, die aus einer Vorverarbeitungsphase (Daten von MMAX2-Annotationsebenen werden aggregiert), der Extraktion der NP-Kandidaten, der Extraktion der NP-Merkmale und der Kandidatenpaare sowie aus einem Resolutionsmodell besteht.
BART verwendet üblicherweise einen auf einem Ansatz von Soon et al. (2001) basierenden Resolutionsalgorithmus, der Kandidaten-NPs hinsichtlich ihrer Merkmale paarweise vergleicht. Statt diesem implementierten wir das Resolutionssystem der Stanford-NLP-Gruppe um Lee et al. (2013), das sich durch seine Sieb-Architektur auszeichnet und regel-basiert ist.
Frühe Systeme zur Koreferenzresolution basierten ebenfalls auf Regeln. Aufgrund ihrer Abhängigkeit von manuell zu bestimmenden Gewichten und ihrer Unfähigkeit, Koreferenten nicht nur paarweise, sondern korpusübergreifend zuzuordnen, werden heute jedoch stattdessen zumeist Machine-Learning-Systeme eingesetzt. Während überwachte Systeme allerdings auf manuell annotierte Daten angewiesen sind, lassen sich unüberwachte Systeme hingegen aufgrund ihrer Komplexität nur schwierig auf neue Domänen übertragen.
Lee et al. (2013) suchen, die Vorteile Regel- und Machine-Learning-basierter Systeme zu verheiraten und konnten so mit ihrem regel-basierten System das beste Ergebnis beim CoNLL-2011 *shared task* erzielen. Im Rahmen der Sieb-Architektur werden nacheinander - absteigend nach ihrer Präzision geordnet - eine Reihe von deterministischen Koreferenzmodellen angewendet, wobei jedes Modell auf den Output seines Vorgängers aufbaut. Besonders das Entität-zentrische Modell, bei dem Merkmale über alle Vorkommen einer Entität geteilt werden, ermöglicht die globale Koreferenzresolution für regel-basierte Systeme.

2 Allgemeine Implementation

Die Klasse `SieveAnnotator` ruft für jedes Dokument die `decodeDocument`-Methode des `SieveDecoder` auf, der `corefResolver` als Interface implementiert. In dieser wird über

alle *mentions* dieses Dokumentes zehnmal iteriert, wobei bei jeder Iteration durch eine `SieveFactory` ein anderer *sieve* aufgerufen wird. Hier filtern wir bereits alle indefiniten *mentions* heraus, da diese sich meist auf generische Konzepte beziehen, die keinen Koreferenten besitzen.
Jeder *sieve* erbt von der abstrakten `Sieve`-Klasse, die `runSieve`-Methode, die er für jede *mention* aufruft und in der er unter den Antezedenzien nach einem Koreferenten für diese *mention* sucht. `Sieve` enthält bereits alle *utility*-Methoden, die von den Subklassen verwendet werden und von denen die meisten als Input eine `PairInstance` bestehend aus *mention* und Antezedens annehmen. Zur Bestimmung mancher Merkmale verwenden wir darüber hinaus die in BART bereits implementierten `PairFeatureExtractors`.
Die Merkmale koreferenter *mentions* werden in einer eigenen `DiscourseEntity`-Klasse geteilt. Weitere sprachspezifische Informationen ergänzten wir in den entsprechenden `LinguisticConstants` sowie durch Methoden zum Nachschlagen in zusammengestellten Listen im korrespondierenden `LanguagePlugin`.

3 Details zu den einzelnen Sieves

- **SpeakerIdentification** identifiziert den Sprecher und verbindet ihn mit möglichen koreferenten Pronomen in wörtlicher Rede. Da der von Lee et al. (2013) verwendete OntoNotes-Korpus auch Telefon-Gespräche und Talk Shows umfasst, erzielt dieser Sieve bereits einen Recall von 8,7% und einen F1-Score von 15,5% (MUC). Da der TüBa-D/Z-Korpus hingegen nur aus Nachrichtenartikeln besteht, ist der Effekt dieses *sieve* signifikant geringer. Handelt es sich bei einem der beiden *mentions* um ein Vorfeld-Es, so wird keine Übereinstimmung gefunden. Numerus-Äquivalenz und eine Satzentfernung ≤ 1 müssen gegeben sein. Es wird mit BARTs `FE_Speech` überprüft, ob sich eine der *mentions* in wörtlicher Rede befindet und auf Grundlage der Position von ':' und '“ sowie Sprechverben ausgemacht, wo sich der Sprecher befindet, woraufhin schließlich sichergestellt wird, dass die sich in der wörtlichen Rede befindende *mention* weder Reflexiv-, noch Relativ-, sondern lediglich Pronomen ist.

- **StringMatch** sieht zwei *mentions* als koreferent an, wenn sie exakt übereinstimmen (einschließlich Modifikatoren und Artikel). Wir überprüfen dies, indem wir den String der *markables* der beiden *mentions* vergleichen. Zudem schließen wir indefinite *mentions* aus sowie *mentions*, deren String eine Datumsangabe enthält, da gleiche Daten in TüBa-D/Z nicht als koreferent gelten. Lee et al. (2013) geben hier eine Präzision von über 90% B^3 an, während wir für das Deutsche und den TüBa-D/Z-Korpus eine etwas geringere Genauigkeit von 86% erreichen.

- **RelaxedStringMatch** gibt zwei *mentions* als koreferent an, sofern sie identisch sind, wenn ihre Postmodifikatoren ignoriert werden.

- **PreciseConstructs** matcht zwei *mentions* als koreferent, wenn sie eine der folgenden Bedingungen erfüllen:

- Sie stehen gemeinsam in einer Appositions- oder Subjekt-Objekt-Konstruktion. Diese Bedingung ist für uns unerheblich, da diese Konstruktionen in TüBa-D/Z nicht annotiert werden.
- Die *mention* modifiziert den Antezedens-Kopf. Hier weisen wir explizit den deutschen Relativpronomen ein Genus zu und überprüfen dessen Übereinstimmung zwischen *mention* und Antezedens sowie Wortentfernung, wobei wir sicherstellen, dass das Antezedens nicht von einer weiteren *mention* eingebettet wird, auf die sich das Relativpronomen eher beziehen könnte.
- Eine der *mentions* ist ein Akronym der anderen. Hier berücksichtigen wir Eigenheiten des Deutschen, in dem auch Akronyme teilweise mit '-' getrennt werden.
- Eine der *mentions* ist ein Demonym der anderen. Hierfür schlagen wir Demonyme in einer auf Wikipedia basierenden Liste nach, wobei dieses Kriterium im TüBa-D/Z-Korpus nicht von großer Bedeutung ist.

Lee et al. (2013) geben für diesen *sieve* eine B^3-Präzision von über 90%, während wir eine Genauigkeit von 81% erhalten. Da das Relativpronomen-Kriterium in diesem *sieve* überwiegt, liegt diese Differenz vermutlich in der hypotaktischen Struktur des Deutschen (im Vergleich zum Englischen), sowie in der domänspezifischen Häufigkeit extraponierter (und daher schwer zuordenbarer) Relativsätze begründet.

- **StrictHeadMatchA** matcht zwei *mentions*, wenn das *head lemma* der Anapher in den *heads* der Antezedens-Entität vorhanden ist und die folgenden Bedingungen erfüllt sind:
  (i) Die Modifikatoren der *mention* müssen unter den Modifikatoren des Antezedens sein.
  (ii) Alle Nicht-Stoppwörter der aktuellen Entität müssen in der Anzedens-Entität vorkommen.
  (iii) Keine der beiden NPs kann Kind des Konstituenten der anderen NP sein (*i-within-i*).

- **StrictHeadMatchB** ignoriert Bedingung (i), während **StrictHeadMatchC** auf Bedingung (ii) verzichtet.

- **ProperHeadNounMatch** weist zwei *mentions* als koreferent aus, wenn sie dasselbe *head word* besitzen und dieses *head word* ein *proper noun* ist, sowie Bedingung (iii) erfüllen und (iv) keine unterschiedlichen Orte, Namen, Zahlen oder räumliche Modifikatoren aufweisen.

- **RelaxedHeadMatch** matcht zwei *mentions*, wenn der Kopf der *mention* mit einem Wort in der Antezedens-Entität übereinstimmt, wobei beide *named entities* desselben Typs sein müssen und Bedingungen (ii) und (iii) genügen müssen.

- **PronounMatch** matcht Pronomen mit ihrem Antezedens, sofern diese morphologisch kompatibel sind, den Binding-Constraints genügen und nicht mehr als drei Sätze auseinanderliegen. Kataphern werden nicht berücksichtigt.
  Das Pronomen ist mit demjenigen Antezedens koreferent, das die höchste *salience* hat (Wunsch 2006). Die *salience* berechnen wir folgendermaßen:

  +20 wenn sich Antezedens und Anapher im selben Satz befinden

  +35 wenn Antezedens und Anapher die selbe grammatische Funktion verwenden

  +170 wenn der Antezedens ein Subjekt ist

  +70 wenn der Antezedens ein Akkusativobjekt ist

  +50 wenn der Antezedens ein Dativ oder Genitivobjekt ist

  +100 wenn der Antezedens ein proper noun ist

  Die endgültige *salience* ist abhängig von der Satzentfernung d: $S = \frac{s}{2^d}$.
  Wir berechnen die *salience* wie Wunsch (2006), berücksichtigen jedoch keine Kataphern und keine Antezedenten, die mehr als 3 Sätze entfernt sind. Außerdem verwenden wir keine die *head noun emphasis*, führen aber zusätzlich *proper noun* als Feature ein.

4 Vergleich der Datenformate und Korpora

Das XML-Format der TüBa-D/Z wird für BART in MMAX2 (Müller und Strube 2006) konvertiert, das die Daten in verschiedenen *markable levels* bereitstellt, wobei auf benötigte Diskurselemente, wie z.B. grammatische Funktionen mit `getDiscourseElementsByLevel` zugegriffen werden kann. Wie bereits eingangs erwähnt unterscheiden sich die Annotationsrichtlinien der verwendeten Korpora und ihre Datenformate hinsichtlich der Konstruktionen, die annotiert werden und ihrer Domäne.

5 Evaluation

Aufgrund der Schwierigkeit, Daten im MMAX2-Format zu analysieren und zu visualisieren, konvertieren wir die Texte in das HTML-Format, wobei wir koreferente *mentions* farbig markieren. Mithilfe der `Evaluation`-Klasse lassen wir uns ebenfalls die Details jedes *match* und die Sieb-spezifische Performanz ausgeben.
Wir evaluierten gegen BART für das Deutsche und gegen das Stanford-System für das Englische. Als Testkorpus für das Deutsche verwendeten wir die ersten 99 Dokumente der TüBa-D/Z, wobei wir BARTs *machine learning*-Komponente auf den restlichen Dokumenten der TüBa-D/Z (Nr. 100 - Nr. 3528) trainierten. In Tabelle 4 ist der Vergleich mit BART zu sehen, während aus Tabelle 2 die Anzahl der verlinkten *mentions* jedes *sieve* aus Tabelle 3 der individuelle Performanzgewinn hervorgehen.

	MUC-Score		
	Recall	**Precision**	**F_1**
Unser System	0.644	0.691	0.667
BART	0.721	0.532	0.612

Tabelle 1: Vergleich mit BARTs Machine Learning-Konfiguration (`XMLExperiment`)

Sieve	**# Links**	**# korrekte Links**	**Präzision**
SpeakerIdentificationSieve	11	7	0.636
StringMatchSieve	324	280	0.864
RelaxedStringMatchSieve	67	44	0.657
PreciseConstructSieve	139	113	0.813
StrictHeadMatchASieve	136	104	0.765
StrictHeadMatchBSieve	182	118	0.648
StrictHeadMatchCSieve	12	6	0.500
ProperHeadNounMatchSieve	2	2	1.000
RelaxedHeadMatchSieve	72	56	0.778
PronounMatchSieve	797	460	0.577

Tabelle 2: Übersicht über die Anzahl der verlinkten *mentions* der *sieves*

	MUC-Score		
	Recall	**Precision**	**F_1**
SpeakerIdentification	0.004	0.637	0.008
+StringMatch	0.157	0.857	0.265
+RelaxedStringMatch	0.180	0.825	0.295
+PreciseConstructs	0.241	0.822	0.372
+HeadMatchA	0.295	0.809	0.432
+HeadMatchB	0.355	0.775	0.487
+HeadMatchC	0.357	0.771	0.488
+ProperHeadNounMatch	0.358	0.771	0.489
+RelaxedHeadMatch	0.383	0.771	0.512
+PronounMatch	0.644	0.691	0.667

Tabelle 3: Performanz der einzelnen *sieves*

Für die Evaluation unseres Systems auf englischsprachigen Daten, die aus Tabelle 4 hervorgeht, verwendeten wir das Trainings-Set des CoNLL-2012 *shared task*, das auf dem OntoNotes 5.0-Korpus basiert. Die Daten wurden mithilfe mehrerer Skripte von Olga Uryupina in das MMAX2-Format konvertiert und außerdem von der BART-eigenen *preprocessing*-Pipeline vorverarbeitet.
Da wir unser System vorrangig für das Deutsche entwickelten, sind diese Ergebnisse deutlich ausbaufähig. Der PronounMatchSieve konnte zudem nicht verwendet werden, da er – anders als der PronounMatchSieve des Stanford-Systems – auf grammatischen Funktionen basiert. Eine Ebene, die diese darstellt, war allerdings nicht verfügbar.

	MUC-Score
	F_1
Unser System	0.420
Stanford	0.603

Tabelle 4: Vergleich mit dem Stanford-System

6 Literatur

Broscheit, S. et al. (2010*a*), BART: A multilingual anaphora resolution system, *in* 'Proceedings of the 5th International Workshop on Semantic Evaluation', SemEval '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 104–107.

Broscheit, S. et al. (2010*b*), Extending BART to provide a coreference resolution system for german., *in* 'LREC', Citeseer.

Lee, H. et al. (2013), 'Deterministic coreference resolution based on entity-centric, precision-ranked rules', *Comput. Linguist.* **39**(4), 885–916.

Müller, C. und Strube, M. (2006), 'Multi-level annotation of linguistic data with MMAX2', *Corpus technology and language pedagogy: New resources, new tools, new methods* **3**, 197–214.

Soon, W. M. et al. (2001), 'A machine learning approach to coreference resolution of noun phrases', *Computational linguistics* **27**(4), 521–544.

Versley, Y. (2006), A constraint-based approach to noun phrase coreference resolution in German newspaper text, *in* 'Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS 2006)'.

Versley, Y. (2011), 'Resolving coreferent bridging in german newspaper text'.

Versley, Y. et al. (2008), BART: A modular toolkit for coreference resolution, *in* 'Proceedings of the ACL-08: HLT Demo Session', Association for Computational Linguistics, Columbus, Ohio, pp. 9–12.

Wunsch, H. (2006), Anaphora resolution–what helps in German, *in* 'Proceedings of the International Conference on Linguistic Evidence', pp. 101–105.

Wunsch, H. (2010), 'Rule-based and memory-based pronoun resolution for German: A comparison and assessment of data sources'.