Linking Entities and Types to Unstructured Text

(Summer School on Natural Language Processing
and Machine Learning)

Soumen Chakrabarti
IIT Bombay
http://www.cse.iitb.ac.in/~soumen/

June 8, 2017

Entity and type linking: Motivation

- Commercial Web search at the outer limits of imprecise search
- SQL, XML, SPARQL, . . . allow precise search
- But demand knowledge of schema, joins, aggregates
- Past and ongoing research: Translating imprecise to precise query
- Limited mileage beyond 1–2 relations
- Can we use a “softer” intermediate query format?
  - Most basic “universal” relations: instance-of, subtype-of
  - Other relations expressed using short text spans
  - Variables, subqueries/clauses, (equi)joins
  - Aggregates (often to establish belief)

Example query

- Let m be a motherboard
- m has two GPU slots
- m supports remote management
- Let c be a dealer
- c sells m
- c has phone number p
- p is listed in Kolkata

Tabulate $\langle m, c, p \rangle$

Pieces to the puzzle

- Recognize that token TZ68K+ corresponds to entity (definition page) https://en.wikipedia.org/wiki/Biostar_TZ68K%2B
- Harder to disambiguate Valley Computer Store
- Most motherboards are out of vocabulary in major knowledge graphs (KGs): DB85FL, S1200V3RPS
- Can still identify them as motherboards from context
- Query planning and evidence aggregation are out of the scope of this tutorial
- Here we focus on entity and type linking

Prerequisites

- Basic text processing: tokenization, stemming, vector space
- Labeling text as a token or span sequence: HMM, CRF, basic inference in graphical models
- Application to coarse named entity recognition (NER): person, location, organization, date, other, ...
- Word embeddings: word2vec, GloVe
- Basic convnet, RNN, LSTM definitions

Distributional vectors and word clusters

- Finite labeled data, feature sparsity, and out-of-vocabulary (OOV) words/features have always troubled POS and NER tagging and estimating n-gram statistics
- E.g. we saw car in the training sequences but never sedan, or Shanghai in test data but only other cities in training data
- But an unlabeled corpus has enough clues that these are related words
- A well-established partial fix is word cluster features
- First proposed by IBM researchers in 1992 http://aclweb.org/anthology/J/J92/J92-4003.pdf
- Given a word like sedan, collect all context windows (say) at most 11 words wide centered on it
- From these contexts collect a bag of other words, count them
- Possibly transform from raw counts to TFIDF

Distributional vectors and word clusters (2)

- Represent as a sparse vector in a space as large as the corpus vocabulary; perhaps scale to unit length
- This is the distributional vector for sedan
- Turns out the d.v.'s of similar/related words are similar
- Can cluster these d.v.'s using standard clustering tools; see https://en.wikipedia.org/wiki/Brown_clustering
- In standard CRF implementations, one of the features for each token is its cluster ID
- The best number of clusters may be application-dependent

Word embeddings

- In a token window, the focus token f is at the center and others are context tokens c
- Each word in the vocabulary is associated with two embeddings, $u_w \in \mathbb{R}^D$ as focus and $v_w \in \mathbb{R}^D$ as context
- Typically D ranges from 100 to 1000
- Two dominant paradigms to train $\boldsymbol{U}, \boldsymbol{V}$

$$\text{GloVe:} \qquad \log X_{fc} \approx u_f \cdot v_c + b_f + b_c,$$

where $b_w \in \mathbb{R}$ is a per-word offset and X_{fc} is the cooccurrence count of words f and c, and

$$\text{Word2vec:} \qquad \Pr(f, c \text{ cooccur}) = \sigma(u_f \cdot v_c),$$

where $\sigma(\bullet) = 1/(1 + e^{-\bullet})$ is the sigmoid function

Word embeddings (2)

- Variations of low-rank factorization of a transformed cooccurrence matrix
- Usually only $\boldsymbol{U}$ used for downstream tasks, one vector per word, usually scaled to unit L2 norm
- Although not explicitly trained to those ends, the focus embeddings are useful for many tasks
  - $u_{\text{auto}} \approx u_{\text{sedan}}$
  - $u_{\text{king}} - u_{\text{man}} + u_{\text{woman}} \approx u_{\text{queen}}$, etc.
- Similarity as dot-product or cosine, vs. dissimilarity as Lp distance
- Sometimes, but not always, interchangeable

Entity embeddings [1, 2]

- Word2vec or GloVe on text corpus will give one embedding per word (bank) or pre-identified compound (Michael_Jordan)
- Even though there are many senses of "bank" and many people called "Michael Jordan"
- Many recent papers on fitting or interpolating embedding per sense rather than token/compound
- Here we begin with the assumption (chicken and egg?) that each mention span has been replaced with a special "word", an entity ID such as `/m/054c1`
- Regard entity IDs as regular words and run word2vec or GloVe
- In Wikipedia, there are at least two forms of association between words and entities
  - The text on the definition page of an entity
  - The text in the context of known ("gold") mentions of an entity

Entity embeddings [1, 2] (2)

- These are expected to follow somewhat different language models
- Combining info from them may lead to better entity representations
- If entities are points, then types are . . . ?
- Extensive literature on how relation triples are represented in vector/matrix/tensor space — dozens of models over the last three years

From coarse NER to fine types

FIGER type catalog (112 fine types)

person	doctor
actor	engineer
architect	monarch
artist	musician
athlete	politician
author	religious_leader
coach	soldier
director	terrorist

organization	terrorist_organization
airline	government_agency
company	government
educational_institution	political_party
fraternity_sorority	educational_department
sports_league	military
sports_team	news_agency

location	body_of_water
city	island
country	mountain
county	glacier
province	astral_body
railway	cemetery
road	park
bridge	

product	camera
engine	mobile_phone
airplane	computer
car	software
ship	game
spacecraft	instrument
train	weapon

art	written_work
film	newspaper
play	music

event	military_conflict
attack	natural_disaster
election	sports_event
protest	terrorist_attack

building	time	chemical_thing	website
airport	color	biological_thing	broadcast_network
dam	award	medical_treatment	broadcast_program
hospital	educational_degree	disease	tv_channel
hotel	title	symptom	currency
library	law	drug	stock_exchange
power_station	ethnicity	body_part	algorithm
restaurant	language	living_thing	programming_language
sports_facility	religion	animal	transit_system
theater	god	food	transit_line

Fine type tagging: Motivation

- Suppose John Smith is a cricket player not yet in Wikipedia
- But mentioned in local news about county cricket
- Query is "Who took four wickets in one over last year against Birmingham?"
- Potential evidence passage[1] is "Birmingham crashed out of the match after losing four wickets to Smith in a single over last month."
- Goal is to collect John Smith as a (strong) candidate, for which we must know that Smith refers to a cricketer[2]
- Experience suggests (thousands of) finer types better for QA than (hundreds of) fine types, but hard to infer from context

[1] Would be very nice to also collect evidence of four wickets from "Alan and Boyd were bowled out by the first two balls from Smith; Ray and Tony were caught out before the over was done."

[2] Must also know that who is asking for a cricketer, not, e.g., a politician, a process called answer/target type inference.

Type tagging: basic idea

- Efficiently produce training data: text with entity mention spans marked out, with type(s) of entities provided as labels
  - Nobody scored as many goals in one match as Messi in 2004.
  - Type of Messi is `/person/athlete`
- Source: Wikipedia links to other Wikipedia pages corresponding to entities
- Collect features from mention context
  - scored, goals, match
- Find types to which these entities belong — these are labels
- (Caveat: Not all these types may be active in a mention context)
- Train a multi-class, multi-label classifier
- At test time, use a B-I-O CRF to locate mention segments
- For each mention, collect features from context
- Predict one or more types using multi-class, multi-label classifier

FIGER system and features

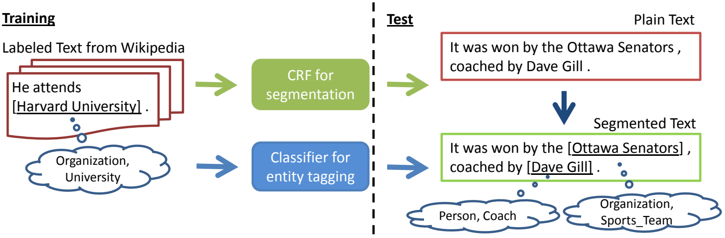


Figure 1: System architecture of FIGER.

Feature	Decription	Example
Tokens	The tokens of the segment.	"Eton"
Word Shape	The word shape of the tokens in the segment.	"Aa" for "Eton" and "A0" for "CS446".
Part-of-Speech tags	The part-of-speech tags of the segment.	"NNP"
Length	The length of the segment.	1
Contextual unigrams	The tokens in a contextual window of the segment.	"victory", "for", "."
Contextual bigrams	The contextual bigrams including the segment.	"victory for", "for Eton" and "Eton ."
Brown clusters	The cluster id of each token in the segment (using the first 4, 8 and 12-bit prefixes).	"4_1110", "8_11100111", etc.
Head of the segment	The head of the segment following the rules by Collins (1999).	"HEAD_Eton"
Dependency	The Stanford syntactic dependency (De Marneffe, MacCartney, and Manning 2006) involving the head of the segment.	"prep_for:seal:dep"
ReVerb patterns	The frequent lexical patterns as meaningful predicates	"seal_victory_for:dep"

Google fine types and baseline system

PERSON

- **artist**
  - actor
  - author
  - director
  - music
- **education**
  - student
  - teacher
- **athlete**
- **business**
- **coach**
- **doctor**
- **legal**
- **military**
- **political figure**
- **religious leader**
- **title**

LOCATION

- **structure**
  - airport
  - government
  - hospital
  - hotel
  - restaurant
  - sports facility
  - theatre
- **geography**
  - body of water
  - island
  - mountain
- **transit**
  - bridge
  - railway
  - road
- **celestial**
- **city**
- **country**
- **park**

ORGANIZATION

- **company**
  - broadcast
  - news
- **education**
- **government**
- **military**
- **music**
- **political party**
- **sports league**
- **sports team**
- **stock exchange**
- **transit**

OTHER

- **art**
  - broadcast
  - film
  - music
  - stage
  - writing
- **event**
  - accident
  - election
  - holiday
  - natural disaster
  - protest
  - sports event
  - violent conflict
- **health**
  - malady
  - treatment
- **award**
- **body part**
- **currency**
- **language**
  - programming language
- **living thing**
  - animal
- **product**
  - camera
  - car
  - computer
  - mobile phone
  - software
  - weapon
- **food**
- **heritage**
- **internet**
- **legal**
- **religion**
- **scientific**
- **sports & leisure**
- **supernatural**

- Minor tweaks to FIGER types
- Improvements in collecting labeled data

Google fine types and baseline system (2)

- Enhanced classification
- Training data expected to have extraneous labels
  - Entity Obama is-a politician, (ex-) POTUS, lawyer, book author, parent, ...
  - In a given context, one or few types may be 'active'
  - But training instance produced with all type labels
- To mitigate problems from extraneous labels, use weighted approximate rank pairwise (WARP) loss
- Features[3] (e.g. for "... who Barack H. Obama first picked ...")

Feature	Description	Example
Head	The syntactic head of the mention phrase	"Obama"
Non-head	Each non-head word in the mention phrase	"Barack", "H."
Cluster	Word cluster id for the head word	"59"
Characters	Each character trigram in the mention head	":ob", "oba", "bam", "ama", "ma:"
Shape	The word shape of the words in the mention phrase	"Aa A. Aa"
Role	Dependency label on the mention head	"subj"
Context	Words before and after the mention phrase	"B:who", "A:first"
Parent	The head's lexical parent in the dependency tree	"picked"
Topic	The most likely topic label for the document	"politics"

[3] "Washington sat on his favorite Barcelona and opened a Newcastle."

Embedding type labels with WARP loss

- Mention contexts represented as x
- A common situation is $x \in \mathbb{R}^D$, for which we choose embedding $f(x) = \boldsymbol{A}x \in \mathbb{R}^H$, where $\boldsymbol{A} \in \mathbb{R}^{H \times D}$
- Want to exploit related types by embedding each type to a vector; similar types expected to embed to similar vectors
- Let $\boldsymbol{\delta}_t$ is the 1-hot vector for t
- Let the tth column of matrix $\boldsymbol{B} \in \mathbb{R}^{H \times T}$ represent the H-dimensional embedding of type t
- I.e., we can use notation $g(t) = \boldsymbol{B}\boldsymbol{\delta}_t$ as the embedding $g(t) \in \mathbb{R}^H$
- The score of a single type label t for context x is $s_t(x) = f(x) \cdot g(\boldsymbol{\delta}_t)$
- Multiple type labels may be valid in both train and test instances

Embedding type labels with WARP loss (2)

- ith labeled instance is $(x_i, \boldsymbol{y}_i)$ where $\boldsymbol{y}_i$ represents a label set, possibly as a few-hot vector in $\{0,1\}^T$
- Exact inference must explore all 2^T label subsets: $\hat{\boldsymbol{y}} = \operatorname{argmax}_{\boldsymbol{y}} f(x) \cdot g(\boldsymbol{y})$
- To avoid high inference cost, cast as label ranking
- Overall score vector $\boldsymbol{s}(x) = (\ldots, s_t(x), \ldots) \in \mathbb{R}^T$
- Goal is to rank all correct labels before any incorrect one
- Loss on instance $x_i, \boldsymbol{y}_i$ is some function of the rank(s) of the correct label(s) in list of types sorted by decreasing score
- Let $\operatorname{rank}(t, \boldsymbol{s}(x))$ be the rank of label t in sorted list

$$\operatorname{rank}(t, \boldsymbol{s}(x)) = \sum_{y' \neq y} \mathbb{I}(s_{y'}(x) \geq s_y(x))$$

- For a single correct t, we can minimize the above rank

Embedding type labels with WARP loss (3)

- For multiple correct ts, there are various options to combine their ranks, e.g., sum
- For instance $x_i, \boldsymbol{y}_i$, consider good type $t \in \boldsymbol{y}_i$, bad type $t' \notin \boldsymbol{y}_i$
- RankSVM loss for such a pair would be $\max\{0, 1 + s_{t'}(x) - s_t(x)\}$
- To incorporate the rank signal of t, define overall WARP loss

$$\sum_{t \in \boldsymbol{y}_i} \sum_{\bar{t} \notin \boldsymbol{y}_i} \mathcal{R}(\operatorname{rank}(t, \boldsymbol{s}(x)) \max\{0, 1 + s_{t'}(x) - s_t(x)\}$$

- Here $\mathcal{R}$ transforms rank into weight; for precision at k, we can use $\mathcal{R} = \sum_{1 \le i \le k} 1/i$
- Not convex

Kernel WSABIE

- Earlier, $s_t(x) = (Ax) \cdot (B\boldsymbol{\delta}_t) = x^\top (A^\top B)\, \boldsymbol{\delta}_t$
- Where $Ax \in \mathbb{R}^H$ and $B\boldsymbol{\delta}_t \in \mathbb{R}^H$
- A and B appear in only the form $A^\top B \in \mathbb{R}^{D \times T}$, but it is constrained to have rank at most H as a form of regularization
- Despite this, observed noisy "fill" in this matrix while training
- Let $P \circ Q$ be the elementwise product of two matrices, i.e., $(P \circ Q)[d,t] = P[d,t]\, Q[d,t]$
- Google system uses $K \in \{0,1\}^{D \times T}$ as a feature selection or additional noise reduction mechanism

$$s_t(x) = x^\top \big(K \circ (A^\top B)\big)\, \boldsymbol{\delta}_t$$

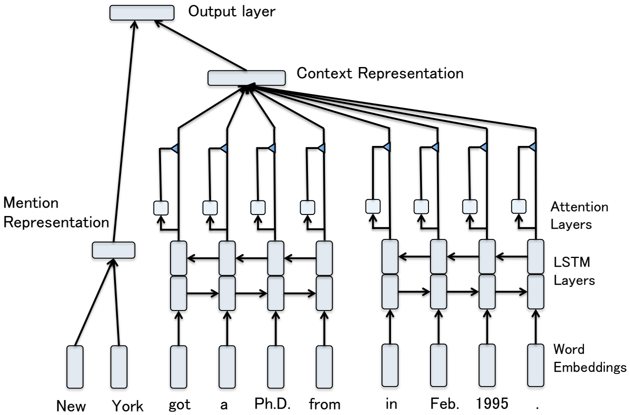
- If $A[:,d]$ is among the 200 nearest neighbors of $B[:,t]$, set $K[d,t] = 1$, and 0 otherwise
- K updated after every iteration (mini-batch?)

Google fine-type system #2 performance

Method	P	R	F1
Ling and Weld (2012)	–	–	69.30
WSABIE	81.85	63.75	71.68
K-WSABIE	**82.23**	**64.55**	**72.35**

Table 4: Precision (P), Recall (R), and F1-score on the FIGER dataset for three competing models. We took the F1 score from Ling and Weld's best result (no precision and recall numbers were reported). The improvements for WSABIE and K-WSABIE over the baseline are statistically significant ($p < 0.01$).

Bi-LSTM fine-type tagger



She got a Ph.D from New York in Feb. 1995.

- Bi-LSTM on left and right context
- Average of word vectors of mention
- +Attention

Bi-LSTM fine-type tagger details

- Let mention words be $M = \{m\}$ with corresponding pretrained (focus) word vectors $u(m)$ from word2vec or GloVe
- Mention vector is designed as $v_m = (1/|M|) \sum_{m \in M} u(m)$
- Suppose we take C words of context from left and right
- Rightmost state from left context LSTM is $\overrightarrow{h}^{\ell}_{C}$
- Leftmost state from right context LSTM is $\overleftarrow{h}^{r}_{1}$
- Context vector is designed as $v_c = \begin{bmatrix} \overrightarrow{h}^{\ell}_{C} \\ \overleftarrow{h}^{r}_{1} \end{bmatrix}$
- Each type t is predicted with

$$\Pr(t|\text{mention, context}) = \sigma\left(W_t \begin{bmatrix} v_m \\ v_c \end{bmatrix}\right)$$

Computing v_c with attention

Sentence	Prediction
... ... The film is a remake of [Secrets (1924)] , a silent film starring Norma Talmadge	/film 0.986 /art 0.982
The film is a remake of Secrets (1924) , a silent film starring [Norma Talmadge]	/person 0.999 /actor 0.987
... ... The festival brought together the foremost filmmakers , including Francois Truffaut , [Roman Polanski] , Robert Enrico , and others	/person 1.00 /director 0.963 /author 0.958 /artist 0.950 /actor 0.871
... ... Jim Hodges , the Democratic nominee , handily defeated Republican Governor [David Beasley] to become the 114th governor of South Carolina	/person 1.00 /politician 0.983
She is best known for roles in various TV Dramas and tokusatsu shows such as [Ultraseven X] and Kamen Rider Kiva	/broadcats_program 0.892

$$e_i^\ell = \tanh\left(W_e \begin{bmatrix} \overrightarrow{h}_i^\ell \\ \overleftarrow{h}_i^\ell \end{bmatrix}\right)$$ L-R & R-L states from left context

$$e_i^r = \cdots$$ L-R & R-L states from right context

$$\tilde{a}_i^\ell = \exp\left(W_a e_i^\ell\right)$$ Attend to important context words

$$\tilde{a}_i^r = \cdots$$

$$a_i^\ell = \frac{\tilde{a}_i^\ell}{\sum_{i=1}^{C}(\tilde{a}_i^\ell + \tilde{a}_i^r)}$$ Normalize attention over left context

$$v_c = \sum_{i=1}^{C} a_i^\ell \begin{bmatrix} \overrightarrow{h}_i^\ell \\ \overleftarrow{h}_i^\ell \end{bmatrix} + a_i^r \begin{bmatrix} \overrightarrow{h}_i^r \\ \overleftarrow{h}_i^r \end{bmatrix}$$ Redefined context representation

LSTM and attention results

Models	P	R	F1
Ling and Weld (2012)	-	-	69.30
Yogatama et al. (2015)	**82.23**	64.55	72.35
Averaging Encoder	68.63	69.07	68.65
LSTM Encoder	72.32	70.36	71.34
Attentive Encoder	73.63	**76.29**	**74.94**

Table 1: Loose Micro Precision (P), Recall (R), and F1-score on the test set

Models	Strict	Loose Macro	Loose Micro
Ling and Weld (2012)	52.30	69.90	69.30
Yogatama et al. (2015)	-	-	72.25
Averaging Encoder	51.89	72.24	68.65
LSTM Encoder	55.60	73.95	71.34
Attentive Encoder	**58.97**	**77.96**	**74.94**

Table 2: Strict, Loose Macro and Loose Micro F1-scores

Reducing (type) label noise [7]

- Fine type training data in the form of spans directly gold-labeled with types is rare
- Wikipedia has millions of pages of text with gold mentions of entities
- Wikipedia, DBpedia, Freebase, WikiData, … have type hierarchies from which we can get all types that contain an entity
- However, most of these types are not relevant at any given mention of the entity
- Training all these types using this textual context would pollute the type models
- Notation: entity e, with mention contexts $C_e = \{c_{ei}\}$ (if e is understood, will drop it)
- e is a member of types in T_e, specified by KG
- I.e., each e associated with $\boldsymbol{y}_e$, a few-hot vector of types

Reducing (type) label noise [7] (2)

- Less realistic to assume per-context gold labels (except to eval fine-type system)
- Each mention context is an instance
- I.e., each entity is associated with multiple instances
- In general each entity has multiple valid labels (types)
- Therefore, a multi-instance multi-label (MIML) setting
- Each context associate with ______ (one/more) types?

MIML approach to fine typing

- Each context c_i will be represented by a fixed-size vector $\boldsymbol{c}_i \in \mathbb{R}^H$ (defined later)
- A first-cut per-mention predictor is a logistic regression: $\Pr(t|c_i) = \sigma(\boldsymbol{w}_t \cdot \boldsymbol{c}_i + b_t)$
- Note multiple t can have score close to 1
- Next, we aggregate in various ways over contexts
- MIML-MAX: Each type $t \in T_e$ is supported by one best context: $\Pr(t|e) = \max_{c \in C_e} \Pr(t|c)$
- Ignores all smaller endorsements
- MIML-AVG: $\Pr(t|e) = \frac{1}{|C_e|} \sum_{c \in C_e} \Pr(t|c)$
- Binary cross entropy $\text{BCE}(y, y') = -y \log y' - (1-y) \log(1-y')$
- All $\boldsymbol{w}_t$s can be trained using cross-entropy loss $L(\{\boldsymbol{w}_t\}) = \sum_e \sum_t \text{BCE}\big(y_{et}, \Pr(t|e; \boldsymbol{w}_t)\big)$

MIML approach to fine typing (2)

- MIML-ATT: Aggregate with attention over contexts
- Apart from $\boldsymbol{w}_t$, associate each t with another vector $\boldsymbol{v}_t$
- Mention contexts of entity e compete for attention: $\alpha_{i,t} = \dfrac{\exp(\boldsymbol{c}_i \cdot \boldsymbol{v}_t)}{\sum_{i'} \exp(\boldsymbol{c}_{i'} \cdot \boldsymbol{v}_t)}$
- Now we build an attention-weighted context representation: $\boldsymbol{a}_t = \sum_i \alpha_{i,t} \boldsymbol{c}_i$
- Use $\boldsymbol{a}_t$ in place of $\boldsymbol{c}_i$ before: $\Pr(t|e) = \sigma(\boldsymbol{w}_t \cdot \boldsymbol{a}_t + b_t)$
- Loss as before
- Additional "deepness": $\alpha_{i,t} = \dfrac{\exp(\boldsymbol{c}_i^\top \boldsymbol{M} \boldsymbol{v}_t)}{\sum_{i'} \exp(\boldsymbol{c}_{i'}^\top \boldsymbol{M} \boldsymbol{v}_t)}$, where $\boldsymbol{M}$ measures the similarity between context and $\boldsymbol{v}_t$

Context representation $\boldsymbol{c}_i$ using convnet

- At the input, read word embeddings
- Apply narrow convnets separately to left and right context of mention to get $\phi_\ell(c), \phi_r(c)$
- Concatenate into $\phi(c)$ and compute $\boldsymbol{c} = \tanh(\boldsymbol{S}\phi(c))$ where $\boldsymbol{S}$ is more model weights
- So overall we have these model weights:
  - Global $\boldsymbol{M}, \boldsymbol{S}$
  - Global weights in convnet ϕ
  - $\boldsymbol{w}_t, \boldsymbol{v}_t, b_t$ for each type
  - Word embeddings (if fine tuned after pretraining)
- Between $\boldsymbol{w}_t, \boldsymbol{v}_t$, is there a usable/interpretable representation of type t?
- (How) do they relate to entity embeddings as in ent2vec?

Noise mitigation results

	$P@1$ all	F_1 all	F_1 head	F_1 tail	MAP
1 MLP	74.3	69.1	74.8	52.5	42.1
2 MLP+MIML-MAX	74.7	59.2	50.7	46.8	41.3
3 MLP+MIML-AVG	77.2	70.6	74.9	56.2	45.0
4 MLP+MIML-MAX-AVG	75.2	71.2	76.4	56.0	47.1
5 MLP+MIML-ATT	81.0	72.0	76.9	59.1	48.8
6 CNN	78.4	72.2	77.3	56.3	47.6
7 CNN+MIML-MAX	78.6	62.2	53.5	49.7	46.6
8 CNN+MIML-AVG	80.8	73.5	77.7	59.2	50.4
9 CNN+MIML-MAX-AVG	79.9	74.3	79.2	59.8	53.3
10 CNN+MIML-ATT	83.4	75.1	79.4	62.2	55.2
11 EntEmb	80.8	73.3	79.9	57.4	56.6
12 FIGMENT	81.6	74.3	80.3	60.1	57.0
13 CNN+MIML-ATT+EntEmb	**85.4**	**78.2**	**83.3**	**66.2**	**64.8**

- ClueWeb with FACC1 entity annotations
- Freebase entities mapped to 102 FIGER types
- 4.3 million contexts
- Head means > 100, tail < 5 mentions

Entity disambiguation

Entity disambiguation

Goal: To refine a mention tag from fine types to the ID of specific entity in a catalog or knowledge graph like Wikipedia or Freebase

- ... book by Mike Jordan on graphical models ...
- ... chance to see Michael Jordan play without Dean Smith ...
- `http://en.wikipedia.org/wiki/Michael_Jordan` or `http://en.wikipedia.org/wiki/Michael_I._Jordan` or ...?
- Which entity catalog to use? (Wikipedia, TAP, OpenCYC, WordNet, ...)
- What about the many Mike Jordans not in the catalog?
- Different from anaphora: Every dog has *its* day

Some distinctions from WSD

- Word sense disambiguation (WSD) is largely about common words, not references to specific entities
  - 42 senses of "run" in WordNet
  - Part of speech helps a fair bit
- Entity catalog typically richer info source than dictionary
  - Broader category system
  - Part of speech is largely "proper noun" and not as helpful
- Entity disambiguation goals:
  - Identify that a sequence of tokens is a potential mention
  - Capture suitable context around to form spot s
  - Assign s to a suitable entity γ in catalog
  - Or claim that there is no suitable γ

Why annotate?

- Make raw text look like Wikipedia with definitional and informational links (most systems)
  - Annotate first occurrence only
  - Annotate only on-topic entities
  - Use discretion to avoid "hyperlink fatigue"
- Index the annotations to enable advanced search (our focus)
  - Exhaustive annotation
  - Make no whole-document topic judgment

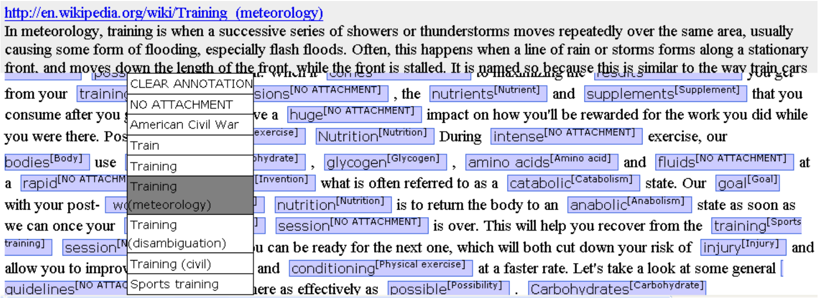
Notation

- book by Mike Jordan on graphical models and chance to see Michael Jordan play without Dean Smith are spots s
- Mike Jordan and Michael Jordan are mentions, other tokens form context
- γ is an entity ID or label from the catalog
- Set S_0 of n spots on page, $s \in S_0$
- Γ_s is the set of labels admissible for spot s
- "No annotation" option $\textsc{na}$
- $y_s \in \Gamma_s \cup \textsc{na}$ is the entity label assigned to spot s
- $\vec{y}$ is a vector of n label variables
- $\Gamma_0 = \bigcup_{s \in S_0} \Gamma_s$

Catalog representation

- Pattern after WordNet, Wikipedia, TAP, ...
- Each entity γ as an associated description
- Descriptions link to other related entities γ'
- Entities belong to one or more categories
- Categories (physicist) are subcategories of others (scientist)
- Links may be "incidental"
- Categories and super-categories may be noisy: *Machine learning researcher* more meaningful than *Living people* or *Year of birth missing*
- Cycles in is-a "hierarchy"?

Human supervision



- System identifies spots and mentions
- Shows pull-down list of (subset of) Γ_s for each s
- User selects $\gamma^* \in \Gamma_s \cup \mathrm{NA}$

SemTag

- Used Stanford TAP ontology (72,000 entities)
- Set of classes C, subclass relation $S \subseteq C \times C$, set of instances (entities) I, many-to-many type relation $T \subseteq I \times C$
- i has class c_1 and c_1 subclass of c_2 implies i has class c_2
- Entity taxonomy is a DAG, $\pi(v)$ is the path up from v to root node r
- Taxonomy node v has label set $L(v)$, e.g., nodes corresponding to cats, football, computers and cars all contain the label 'jaguar'

SemTag output example

```
The <resource ref="http://tap.stanford.edu/BasketballTeam
_Bulls">Chicago Bulls</resource> announced yesterday that
<resource ref="http://tap.stanford.edu/AthleteJordan,_Mich
ael">Michael Jordan</resource> will ...
```

- Functionally identical to inserting Wikipedia links in free-form text
- Wikipedia is more organic than TAP; has poorer quality category hierarchy

SemTag disambiguation

- $\mathrm{sim}(u, s) \in [0, 1]$ is a local similarity between catalog node u and (context of) spot s
- $\mathrm{sim}(\cdot, \cdot) = \frac{1}{2}$ is "most uncertain"
- Node v is eligible for spot s if

$$\text{root } r \neq \arg\max_{u \in \pi(v)} \mathrm{sim}(u, s)$$

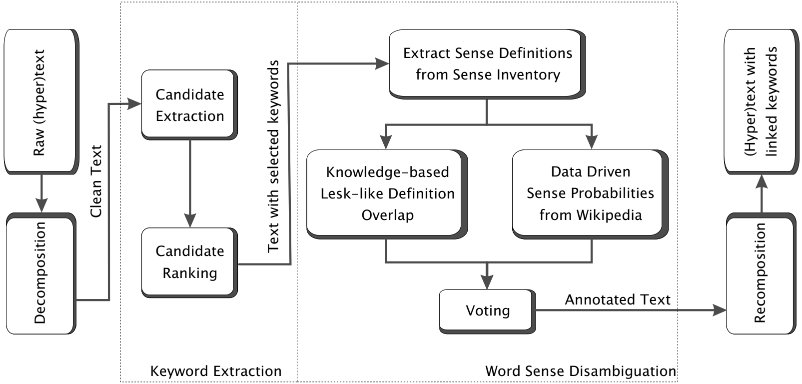
  i.e., some node on $\pi(v)$ other than root most similar to s
- Supplement eligibility with human-judged scores of reliability at each node u
  - m_u^a = probability that spots for subtree rooted at u are "on topic"
  - m_u^s = probability that automatic eligibility judgment is correct

SemTag TBD algorithm

- To decide whether to link spot s to node v ...
- Find nearest ancestor u of v that has human-judged reliability scores
- If $|\frac{1}{2} - m_u^a| > |\frac{1}{2} - m_u^s|$, return $\mathrm{sign}(m_u^a - \frac{1}{2})$
- Else if $m_u^s > \frac{1}{2}$ (eligibility judgment is often correct), return $\mathrm{eligible}(c, u)$
- Else (eligibility judgment is often wrong) return $1 - \mathrm{eligible}(c, u)$

(Can regard as a simple hand-tuned form of stacked learning)

Wikify!



- Two-phase process
- First identify token spans "worthy of annotation"
- Then choose entity labels

Sample annotations

In 1834, Sumner was admitted to the [[bar (law)|bar]] at the age of twenty-three, and entered private practice in Boston.

It is danced in 3/4 time (like most waltzes), with the couple turning approx. 180 degrees every [[bar (music)|bar]].

Vehicles of this type may contain expensive audio players, televisions, video players, and [[bar (counter)|bar]]s, often with refrigerators.

Jenga is a popular beer in the [[bar (establishment)|bar]]s of Thailand

---

This is a disturbance on the water surface of a river or estuary, often cause by the presence of a [[bar (landform)|bar]] or dune on the riverbed.

Choosing token spans to annotate ("spotting")

- Wikify! follows the Wikipedia philosophy
- Use some score to rank candidate spans
- TFIDF of a token in a document
- χ^2 test:

count of token in doc	count of all other tokens in doc
count of token in other docs	count of all other tokens in other docs

- "Keyphraseness" — In how many Wikipedia documents is the same term made a link anchor?
- (They only consider as candidates words which appear at least five times in Wikipedia)

Disambiguation

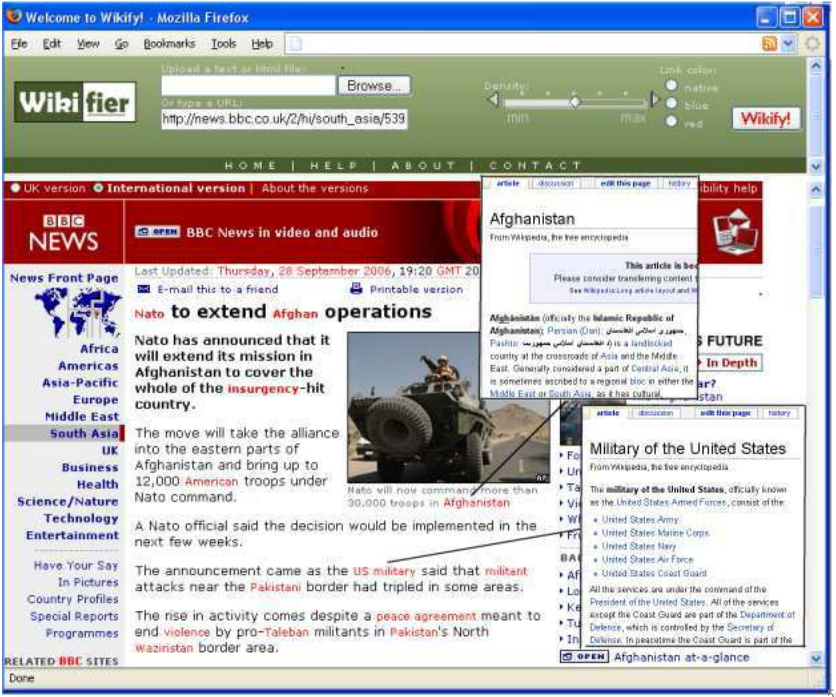
Wikify! compares two local techniques:

- "Knowledge-based approach" — similarity between Wikipedia page text of entity γ and context words in spot s
- "Data-driven approach" — similarity between context of known links to γ and context words in spot s
- "Context" consists of ± 3 words around mention, their parts of speech, salient words chosen from whole document

Results

- "Data-driven" better than "knowledge-based"
- Consensus (agreement) has highest precision

	Words		Evaluation		
Method	(A)	(C)	(P)	(R)	(F)
	Baselines				
Random baseline	6,517	4,161	63.84	56.90	60.17
Most frequent sense	6,517	5,672	87.03	77.57	82.02
	Word sense disambiguation methods				
Knowledge-based	6,517	5,255	80.63	71.86	75.99
Feature-based learning	6,517	6,055	92.91	**83.10**	**87.73**
Combined	5,433	5,125	**94.33**	70.51	80.69


Welcome to Wikify! - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Wikifier
Browse...
http://news.bbc.co.uk/2/hi/south_asia/539
Wikify!
HOME | HELP | ABOUT | CONTACT
UK version International version | About the versions
BBC NEWS
BBC News in video and audio
Last Updated: Thursday, 28 September 2006, 19:20 GMT 20
E-mail this to a friend
Printable version
Nato to extend Afghan operations
Nato has announced that it will extend its mission in Afghanistan to cover the whole of the insurgency-hit country.
The move will take the alliance into the eastern parts of Afghanistan and bring up to 12,000 American troops under Nato command.
Nato will now command more than 30,000 troops in Afghanistan
A Nato official said the decision would be implemented in the next few weeks.
The announcement came as the US military said that militant attacks near the Pakistani border had tripled in some areas.
The rise in activity comes despite a peace agreement meant to end violence by pro-Taleban militants in Pakistan's North Waziristan border area.
News Front Page
Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
Business
Health
Science/Nature
Technology
Entertainment
Have Your Say
In Pictures
Country Profiles
Special Reports
Programmes
RELATED BBC SITES
Afghanistan
From Wikipedia, the free encyclopedia
Afghanistan (officially the Islamic Republic of Afghanistan; Persian (Dari): جمهوری اسلامی افغانستان, Pashto: د افغانستان اسلامي جمهوریت) is a landlocked country at the crossroads of Asia and the Middle East. Generally considered a part of Central Asia, it is sometimes ascribed to a regional bloc in either the Middle East or South Asia, as it has cultural
Military of the United States
From Wikipedia, the free encyclopedia
The military of the United States, officially known as the United States Armed Forces, consist of the
United States Army
United States Marine Corps
United States Navy
United States Air Force
United States Coast Guard
All the services are under the command of the President of the United States. All of the services except the Coast Guard are part of the Department of Defense, which is controlled by the Secretary of Defense. In peacetime the Coast Guard is part of the
FUTURE
In Depth
Afghanistan at-a-glance
Done

Modeling local compatibility

- Feature vector $f_s(\gamma) \in \mathbb{R}^d$ expresses local textual compatibility between (context of) spot s and candidate label γ
- One element of $f_s(\gamma)$ might be the TFIDF cosine similarity between tokens from the context of spot s (say ± 10 tokens) and whole page of description for entity γ
- Another element may be derived of "anchor text" match:
  - Find all links to γ from within Wikipedia
  - Collect anchor text from all these links in a bag of words
  - Find TFIDF cosine similarity between this bag and the spot context s

The sense probability prior

- What entity does "Intel" refer to?
  - Chip design and manufacturing company
  - Fictional cartel in a 1961 BBC TV serial
- $\Pr_0(\gamma|s)$ is very high for chip maker, low for cartel
- Append element $\log \Pr_0(\gamma|s)$ to $f_s(\gamma)$
- "$\log$" will be explained later

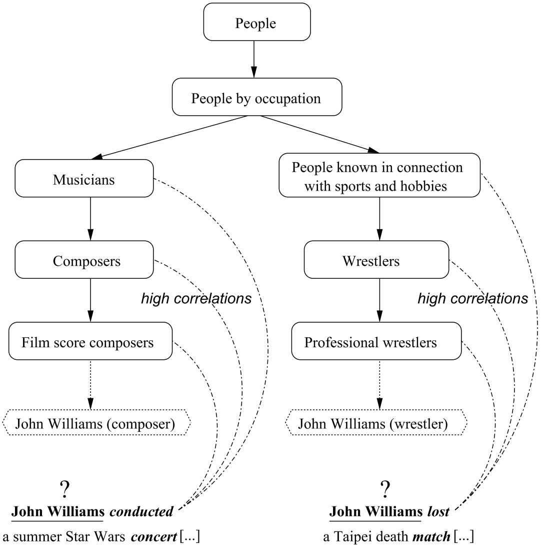
Node score

- Node scoring model $w \in \mathbb{R}^d$
- Node score defined as $w^\top f_s(\gamma)$
- w is trained to give suitable relative weights to different compatibility measures and aggregate the evidence
- During test time, greedy choice local to s would be $\arg\max_{\gamma \in \Gamma_s} w^\top f_s(\gamma)$
- Early algorithms are variations on this theme

Limitations of $\mathrm{sim}(\gamma, s)$

- Training data is sparse
- Direct overlap of words between description of entity γ and context of spot s may be limited
- But overlap between ancestors of γ and context of s may be more reliable

Word-category correlations



Designing tree kernels

- Let $C(\gamma)$ be all ancestor categories of entity γ
- Let $T(s)$ be the text in the context of spot s
- For every word w and every all categories c, define a feature

$$\phi_{w,c}(s,\gamma) = \begin{cases} 1 & \text{if } w \in T(s) \text{ and } c \in C(\gamma) \\ 0 & \text{otherwise} \end{cases}$$

- Run through all possible w, c, e.g., (“conducted”, $musician$), (“concert”, $wrestler$)
- Pad $(\phi_{w,c})$ with local compatibility features
- Finally, get feature vector $\Phi(s,\gamma)$

Learning

- Model as classification: correct/incorrect (s, γ) pair should be labeled $+1/-1$ respectively
- Similar to sequence labeling: $\arg\max_\gamma w^\top \Phi(s, \gamma)$; same max-margin training
- What about spots that do not have any suitable entity in the catalog?
- Out-of-catalog entity $\hat{\gamma}$, with $C(\hat{\gamma}) = \varnothing$ and $T(\hat{\gamma}) = \varnothing$
- One last feature element $\phi_{\hat{}}(s, \gamma) = [\![\gamma = \hat{\gamma}]\!]$
- Equivalent to automatically learning a (lower) threshold on $w^\top \Phi(s, \gamma)$

Tree kernel results

Data set	TreeKernel	TextOnly
People by occupation, top 110	0.772	0.615
Ditto, all 540	0.684	0.558
Ditto, categories with ≥ 20 entities	0.680	0.554

- Summary: tree kernel better than comparing only text

Relatedness info from entity catalog

- How related are two entities γ, γ' in Wikipedia?
- Embed γ in some space using $g : \Gamma \to \mathbb{R}^c$
- Define relatedness $r(\gamma, \gamma') = g(\gamma) \cdot g(\gamma')$ or related
- Cucerzan's proposal: $c =$ number of categories; $g(\gamma)[\tau] = 1$ if γ belongs to category τ, 0 otherwise

$$r(\gamma, \gamma') = \frac{g(\gamma)^\top g(\gamma')}{\sqrt{g(\gamma)^\top g(\gamma)}\sqrt{g(\gamma')^\top g(\gamma')}},$$

(standard cosine)

Relatedness info from entity catalog (2)

- Milne and Witten's proposal: $c =$ number of Wikipedia pages; $g(\gamma)[p] = 1$ if page p links to page γ, 0 otherwise

$$r(\gamma, \gamma') = \frac{\log \dfrac{|g(\gamma) \cap g(\gamma')|}{|g(\gamma) \cup g(\gamma')|}}{\log \dfrac{c}{\min\{|g(\gamma)|, |g(\gamma')|\}}}$$

- Related to Jaccard
- With voice of small inlink sets attenuated

Leave-one-out disambiguation

- Let Γ_0 be all possible entity disambiguations for all spots on a page
- Precompute the average vector $g(\Gamma_0) = \sum_{\gamma \in \Gamma_0} g(\gamma)$
- Score of candidate label γ for spot s depends on two factors multiplied together
- The local compatibility score as before
- $g(\gamma)^\top g(\Gamma_0 \setminus \{\gamma\}) = g(\gamma)^\top \sum_{\gamma' \in \Gamma_0 \setminus \gamma} g(\gamma')$
- Note that $\Gamma_0 \setminus \gamma$ still contains contributions from entities that cannot be used simultaneously to label the page
- $g(\Gamma_0 \setminus \gamma)$ may not be a representative feature vector

Commonness, usefulness, relatedness

Depth-first search

From Wikipedia, the free encyclopedia

Depth-first search (DFS) is an algorithm for traversing or searching a tree, tree structure or graph. One starts at the root (selecting some node as the root in the graph case) and explores as far as possible along each branch before backtracking.

Formally, DFS is an uninformed search that progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found, or until it hits a node that has no children. Then the search backtracks, returning to the most recent node it hadn't finished exploring. In a non-recursive implementation, all freshly expanded nodes are added to a LIFO stack for exploration.

sense	commonness	relatedness
Tree	92.82%	15.97%
Tree (graph theory)	2.94%	59.91%
Tree (data structure)	**2.57%**	**63.26%**
Tree (set theory)	0.15%	34.04%
Phylogenetic tree	0.07%	20.33%
Christmas tree	0.07%	0.0%
Binary tree	0.04%	62.43%
Family tree	0.04%	16.31%
...		

- "Tree" has many senses, common and rare
- But a low probability sense may be the correct one, based on relatedness to unambiguous anchor entities mentioned near "tree"
- Not all anchors equally useful: "until" vs. "LIFO"

Milne and Witten's recipe

- Identify unambiguous spots $S_!$ from all spots S_0
- Denote $\Gamma_! = \bigcup_{s \in S_!} \Gamma_s$, note that $\Gamma_! \xleftrightarrow{1:1} S_!$
- Ambiguous spot $s \mapsto \Gamma_s$, have to pick $\gamma \in \Gamma_s$
- Each candidate γ is scored based on three signals

  Commonness of γ, i.e., sense probability prior $\Pr_0(\gamma|s)$

  Average relatedness to anchor entities $\gamma_!$, weighted by the usefulness $u(\gamma_!)$ of $\gamma_!$

$$\frac{\sum_{\gamma_! \in \Gamma_! \setminus \gamma} u(\gamma_!) r(\gamma, \gamma_!)}{\sum_{\gamma_! \in \Gamma_! \setminus \gamma} u(\gamma_!)}$$

$$\text{where } u(\gamma) = \sum_{\gamma'' \in \Gamma_! \setminus \gamma'} r(\gamma', \gamma'')$$

  Overall context quality for the spot, $\sum_{\gamma_!} u(\gamma_!)$

Milne and Witten's recipe (2)

- These three signals are presented as features to a classifier (bagged decision tree worked best)
- The label is whether γ is correct for s

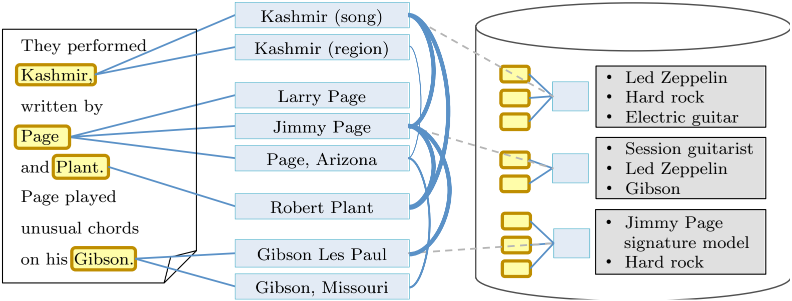
M&W results

	recall	precision	f-measure
Random sense	56.4	50.2	53.1
Most common sense	92.2	89.3	90.7
Medelyan *et al.* (2008)	92.3	93.3	92.9
Most valid sense	95.7	**98.4**	**97.1**
All valid senses	**96.6**	97.0	96.8

- Random sense gives precision over $\frac{1}{2}$, only around two senses per spot
- Recall is as per (reticent) Wikipedia annotation policy

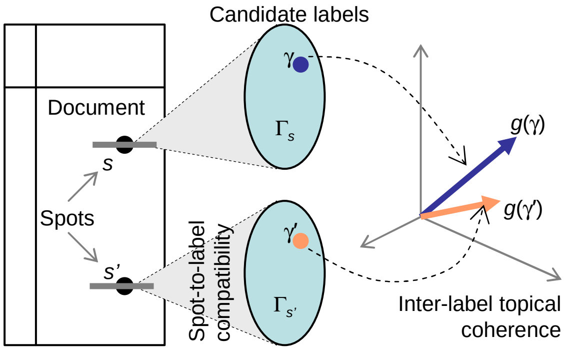
correct	76.4
incorrect (wrong destination)	0.9
incorrect (irrelevant and/or unhelpful)	19.8
incorrect (unknown reason)	2.9

The need for collective disambiguation



- Some entity pairs are more compatible than others
- Compatibility may have different notions (next slide)
- Better to choose per-mention entity labels to maximize pairwise compatibility
- Intractable in general
- Each practical approach sacrifices some aspect to do better in others

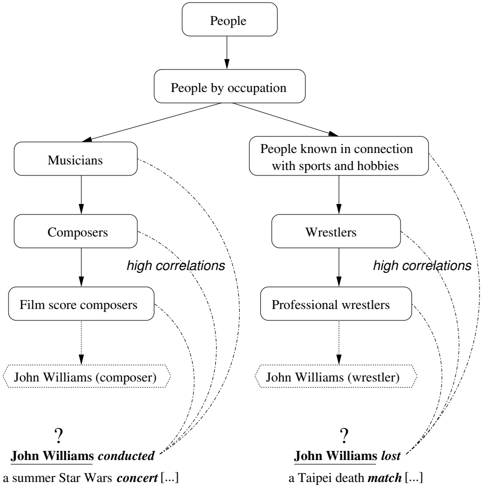
Toward collective formulation



- Premise: coherent doc refers to entities about related categories
- Optimize wrt y an objective with two parts:
  - Local compatibility between s and y_s
  - Global coherence between y_s and $y_{s'}$ for all spot pairs

Local compatibility

- Between mention context and entity
- Entity representations
  - Text on definition pages in Wikipedia
  - Text from gold mention contexts
  - Types that contain the entity
- Between context and types containing entity [10]
- Between page topic/s and entity type/s [12]



Pairwise compatibility

- Entities belong to related types
  - Soccer coaches, clubs, players
- Entities connected by short path in knowledge graph
- Frequently co-cited, with similar embeddings
- $s(A, B) = \dfrac{\log \max(|A|, |B|) - \log(|A \cap B|)}{\log |U| - \log \min(|A|, |B|)}$
- A and B may be pages linking to entities, types containing them, etc.
- Some of these can be coded into $g(\gamma)$
- Others coded as $\Lambda \cdot \phi(\gamma, \gamma')$

Two-part objective to maximize

Node potential:

$$\mathrm{NP}(y) = \prod_s \mathrm{NP}_s(y_s) = \prod_s \exp\left(w^\top f_s(y_s)\right)$$

Clique potential:

$$\mathrm{CP}(y) = \exp\left(\sum_{s \neq s'} g(y_s)^\top g(y_{s'})\right)$$

After taking logs and rescaling terms

$$\frac{1}{|S_0|} \sum_s w^\top f_s(y_s) + \frac{1}{\binom{|S_0|}{2}} \sum_{s \neq s'} g(y_s)^\top g(y_{s'})$$

Two-part objective to maximize

Node potential:

$$\mathrm{NP}(y) = \prod_s \mathrm{NP}_s(y_s) = \prod_s \exp\left(w^\top f_s(y_s)\right)$$

Clique potential:

$$\mathrm{CP}(y) = \exp\left(\sum_{s \neq s'} g(y_s)^\top g(y_{s'})\right)$$

After taking logs and rescaling terms

$$\frac{1}{|S_0|} \sum_s w^\top f_s(y_s) + \frac{1}{\binom{|S_0|}{2}} \sum_{s \neq s'} g(y_s)^\top g(y_{s'})$$

Probabilistic interpretation

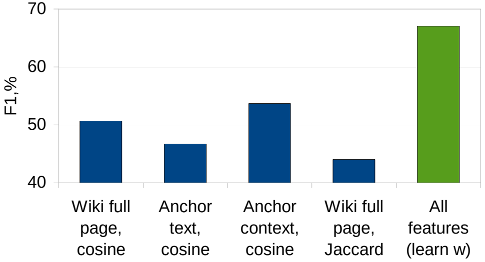
$$\Pr(\vec{y}|\vec{s}) \propto \mathrm{CP}(\vec{y})\, \mathrm{NP}_{\vec{s}}(\vec{y})$$

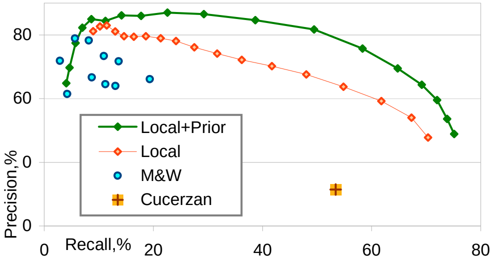
$$\Pr(\vec{y}|\vec{s}) = \frac{1}{Z(\vec{s})} \left(\prod_{s \neq s'} \exp\left(g(y_s)^\top g(y_{s'}) \right) \right) \left(\prod_{s} \exp\left(w^\top f_s(y_s) \right) \right)$$

where $$Z(\vec{s}) = \sum_{\vec{y}} \left(\prod_{s \neq s'} \exp\left(g(y_s)^\top g(y_{s'}) \right) \right) \left(\prod_{s} \exp\left(w^\top f_s(y_s) \right) \right)$$

- (Conditional) probabilistic graphical model with complete graph
- Aka the quadratic assignment problem
- Notoriously difficult NP-hard problem
- Local hill-climbing, LP relaxations

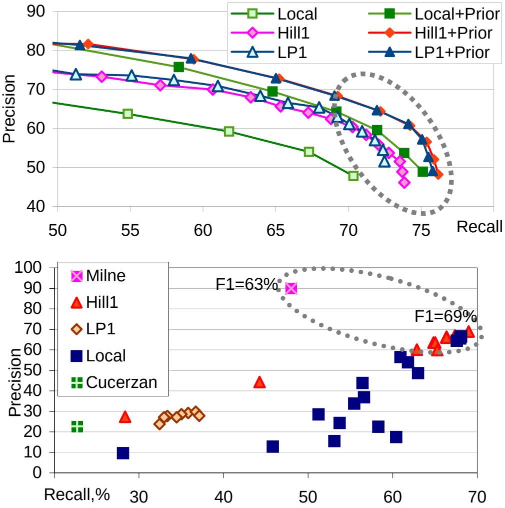
Effect of NP learning





- Learning w is better than commonly-used single features
- Enough to beat leave-one-out and anchor-based approaches

Benefits of collective labeling



- Two different data sets (Web, newswire)
- Can significantly push recall while preserving precision

Trouble with all-pairs clique potential

- Entities in doc may not all be in one cluster
- KG may not know of common type-to-type relations, e.g., cricketers and business tycoons, or politicians and real estate barons
- Less salient entities may not find enough support from other spots
- Asserting all-pairs potentials across coherent clusters needlessly adds noise floor to objective
- Discussed by Kulkarni et al. [13] but not addressed

Single link baseline

- As an extreme simplication of the clique potential, for each mention, find one best supporter

$$g_{\mathsf{SL}}(\boldsymbol{y}) = \prod_i s_i(y_i) \left[\max_j s_{ij}(y_i, y_j) \right]$$

- $\boldsymbol{y}$ is the vector of entity labels assigned to all mentions in a document
- $s_i(y_i)$ is the local score for entity label y_i for mention/spot i
- MAP inference is still intractable
  - If j is the best supporter of i, is i necessarily the best supporter of j?
- Approximate by message passing (loopy belief propagation) on factor graph
- Factor a_i for each mention i

Single link baseline (2)

- Each factor connects to all (mention) nodes, but best supporter makes message passing practical
- Message from a_k to mention i is

$$n_{a_k \to i}(y_i) = \max_{\boldsymbol{y}_{\setminus i}} \Big[\psi_k(y_i, \boldsymbol{y}_{\setminus i}) \prod_{j \neq i} m_{j \to a_k}(y_j) \Big]$$

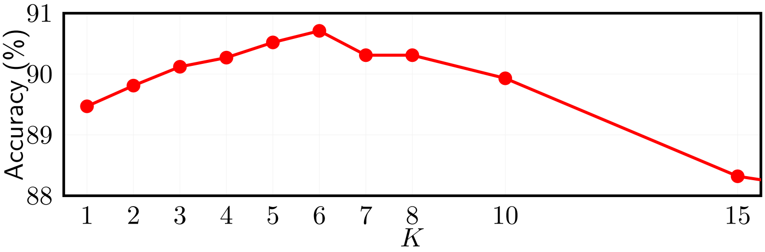
- Belief in $\boldsymbol{y}$ based on incoming messages from all factors

Relaxing to a star model

- Give up global consistency for tractability
- In turn, make each mention center of a star
- Assign label to each spoke separately to maximize support for hub
- Support for label y_i from mention j is $q_{ij}(y_i) = \max_{y_j}[s_{ij}(y_i, y_j) + s_j(y_j)]$
- Score function for mention i is $f_i(y_i) = s_i(y_i) + \sum_{\text{all } j \neq i} q_{ij}(y_i)$
- Predict y_i by maximizing above score
- Next step: replace all $j \neq i$ with something more robust
- In what follows, let $\boldsymbol{q}_i(y_i) = \langle q_{i1}(y_i), \ldots, q_{in}(y_i) \rangle$ be the sequence of support from other mentions to mention i

Multifocal attention (or, you only need six friends)

- Instead of seeking support from all other mentions . . .
- For a nonnegative sequence z, let $\text{amx}_K(z)$ be the sum of the largest K elements of z
- Redefine score function for ith mention as $f_i(y_i) = s_i(y_i) + \text{amx}_K(\boldsymbol{q}_i(y_i))$
- If the document has n mentions with C candidates per mention, inference now takes $O(nC^2 + n \log n)$ time
- K mentions get full attention, others get 0



The last step: from max to soft-max

Using entity embeddings: Three-part optimization

- Overall likelihood fitted through simultaneous maximization
  $\mathcal{L} = \mathcal{L}_w + \mathcal{L}_e + \mathcal{L}_a$
  Word-word: $\mathcal{L}_w$, standard word2vec on text corpus
  Entity-entity: $\mathcal{L}_e$, as expressed through KG
  Word-entity: $\mathcal{L}_a$, connecting mention context words and entity embeddings
- e, e' are related if there is a link between them in the KG, and $e \neq e'$, in which case we want large

$$\mathcal{L}_e = \sum_{e,e'} \log \Pr(e'|e), \qquad \text{where}$$

$$\Pr(e'|e) = \frac{\exp(\boldsymbol{u}_e \cdot \boldsymbol{v}_{e'})}{\sum_e \exp(\boldsymbol{u}_e \cdot \boldsymbol{v}_e)}$$

Using entity embeddings: Three-part optimization (2)

- As in skip-gram, predict mention context words given focus entity ID
- Let M_e be mentions of entity e, $m \in M_e$ be one mention, and $w \in m$ a mention word

$$\mathcal{L}_a = \sum_e \sum_{m \in M_e} \sum_{w \in m} \log \Pr(w|e),$$

where $$\Pr(w|e) = \frac{\exp(\boldsymbol{v}_w \cdot \boldsymbol{u}_e)}{\sum_{w'} \exp(\boldsymbol{v}_{w'} \cdot \boldsymbol{u}_e)}$$

- As is common, softmax is replaced by negative samples

Inference with coherence

- Given a document with many mention spots
- For each mention, compute context vector as average of neighboring word vectors
- (Nothing more fancy like convnet or RNN)
- Set initial entity labels using cosine with context vectors
- Now define the coherence of an entity with others as average cosine between entity vectors
- Reassign most coherent label in a second step
- Crude two-step loopy BP?

Joint word-entity embeddings: NED results

	CoNLL (Micro)	CoNLL (Macro)	TAC10 (Micro)
Our Method	**93.1**	**92.6**	**85.2**
Hoffart et al., 2011	82.5	81.7	-
He et al., 2013	85.6	84.0	81.0
Chisholm & Hachey, 2015	88.7	-	80.7
Pershina et al., 2015	91.8	89.9	-

The sad tale of tail entities

https://en.wikipedia.org/wiki/Michael-Hakim_Jordan:

> *"For most of his life he was known as Michael Jordan, but since he is not related to the more prominent American basketball player of the same name, and got tired of the constant comparisons, he included his second name to his title, thus he became also referred to as Michael-Hakim Jordan."*

- How do humans identify tail entities?
- Highly selective attention to context and other entities

References

I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," *arXiv preprint arXiv:1601.01343*, 2016.
https://arxiv.org/pdf/1601.01343.pdf

O.-E. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," *arXiv preprint arXiv:1704.04920*, 2017.
https://arxiv.org/pdf/1704.04920.pdf

X. Ling and D. S. Weld, "Fine-grained entity recognition." in *AAAI*, 2012.
http://xiaoling.github.io/pubs/ling-aaai12.pdf

References (2)

- D. Gillick, N. Lazic, K. Ganchev, J. Kirchner, and D. Huynh, "Context-dependent fine-grained entity type tagging," *arXiv preprint arXiv:1412.1820*, 2014.
  https://arxiv.org/pdf/1412.1820.pdf
- D. Yogatama, D. Gillick, and N. Lazic, "Embedding methods for fine grained entity type classification," in *ACL Conference*, 2015, pp. 26–31.
  http://anthology.aclweb.org/P/P15/P15-2048.pdf
- S. Shimaoka, P. Stenetorp, K. Inui, and S. Riedel, "An attentive neural architecture for fine-grained entity type classification," *arXiv preprint arXiv:1604.05525*, 2016.
  https://arxiv.org/pdf/1604.05525.pdf

References (3)

- Y. Yaghoobzadeh, H. Adel, and H. Schütze, "Noise mitigation for neural entity typing and relation extraction," *arXiv preprint arXiv:1612.07495*, 2016. https://arxiv.org/pdf/1612.07495.pdf
- S. Dill *et al.*, "SemTag and Seeker: Bootstrapping the semantic Web via automated semantic annotation," in *WWW Conference*, 2003, pp. 178–186.
- R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *CIKM*, 2007, pp. 233–242. http://portal.acm.org/citation.cfm?id=1321440.1321475
- R. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *EACL*, 2006, pp. 9–16. http://www.cs.utexas.edu/~ml/papers/encyc-eacl-06.pdf

References (4)

S. Cucerzan, "Large-scale named entity disambiguation based on Wikipedia data," in *EMNLP Conference*, 2007, pp. 708–716.
http://www.aclweb.org/anthology/D/D07/D07-1074

J. Hoffart *et al.*, "Robust disambiguation of named entities in text," in *EMNLP Conference*. Edinburgh, Scotland, UK: SIGDAT, Jul. 2011, pp. 782–792.
http://aclweb.org/anthology/D/D11/D11-1072.pdf

S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of Wikipedia entities in Web text," in *SIGKDD Conference*, 2009, pp. 457–466.
http://www.cse.iitb.ac.in/~soumen/doc/CSAW/