NED

Collective Named Entity Disambiguation via Personalized Page Rank and Context Embeddings

CS 273 · DATA AND KNOWLEDGE BASES · TERM PROJECT

BY LUIS ÁNGEL

PROBLEM STATEMENT

Given a document d with M named entity mentions $\{m_1, m_2, \dots, m_M\}$, find their **best** (real-world, surrogate) mapping entities $E = \{e_1, e_2, \dots, e_M\}$ in a knowledge base.

a) Exploit the **local features** of entity mentions.

b) Take advantage of the **semantic relatedness** of true entities in d.

Note: NED is an orthogonal task to *Named Entity Recognition*. We work on a set of given annotated entity mentions in text as input to our system.

BASIS AND CONTRIBUTIONS

- This work is based on the foundations provided in:

  [1] *Graph Ranking for Collective Named Entity Disambiguation*, by Alhelbawy, A., and Gaizauskas, R. 2014.

  [2] *Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling*, by Shen, W., Wang, J., Luo, P., and Wang, M. 2013.

  [3] *A Simple but Though-to-Beat Baseline for Sentence Embeddings*, by Arora, S., Liang, Y., and Ma, T. 2017.

We have combined the **sentence embeddings** approach to formalize a local context feature metric for named entity mentions with the topical relatedness expressed through **Wikilinks** to collaboratively resolve the NED problem using **personalized PageRank** with **maximal discriminant** selection

EXAMPLE

Thrown into the middle of [[**Madonna**]]'s whirlwind, [[**Johan Renck**]] had to hit the ground running, just like many of the dancers cast for the clip. [[**Madonna**]] wanted to use a few performers from her tour, such as [[**Daniel Campos**]], Miss Prissy from [[**LaChapelle**]]'s "[[**Rize**]]" crew and traceur [[**Sebastien Foucan**]], a practitioner of parkour, a philosophical French sport that involves moving via uninterrupted motion, whether over, under, through or around objects. "It's not about the music, but the bodily expression," [[**Johan Renck**]] said. "We wanted to show the whole spectrum, be it krumping, breakdancing, jazz or disco."

*Excerpt from **MTV News***
Source: http://www.mtv.com/news/1539338/with-no-director-and-broken-ribs-madonna-was-hung-up-vmas-behind-the-camera/

Madonna

Madonna (entertainer); Madonna (art); Mary (mother of Jesus), Madonna (studio)…

Johan Renck

Bo Johan Renck

Daniel Campos

Daniel Campos Province; Bolivia; ***Cloud (dancer)***; …

LaChapelle

David LaChapelle; Lachapelle, Tarn-et-Garonne

Rize

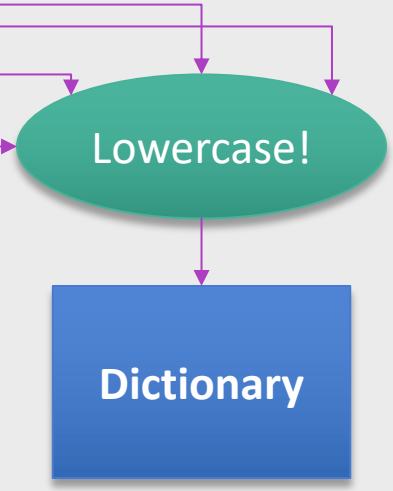
Rize; Rize (band); Rize Province; ***Rize (film)***; …

Sebastien Foucan

Sébastien Foucan

METHODOLOGY · CANDIDATE GENERATION[1]

- For each $m_i \in M$, its mapping entity e_i should be the entity referred to by the **surface form** m_i in d.
- We need a set of **candidate mapping entities**, R_i, such that e_i corresponds to some $r_{i,j} \in R_i$.
- Create a dictionary from **English Wikipedia**, using:
  - **Entity page**: "**Madonna (entertainer)**" becomes *Madonna*.
  - **Redirect page**: "**Sébastien Foucan**" is redirected from *Sebastien Foucan*.
  - **Internal Wikilinks**: "**David LaChapelle**" is linked from [[David LaChapelle|LaChapelle]].
  - **Disambiguation page**: "**Rize (film)**" is listed in Rize (disambiguation).
- And count how many times each $r_{i,j}$ is referred to by the surface form m_i.



METHODOLOGY · CANDIDATE GENERATION[2]

A portion of the dictionary

Surface form	Candidates	Count
madonna	**Madonna (entertainer)** Madonna (art) Mary (mother of Jesus) [16 more candidates]	4805 178 55 …
johan renck	**Bo Johan Renck**	1
lachapelle	**David LaChapelle** Lachapelle, Tarn-et-Garonne	1 1
daniel campos	Daniel Campos Province Bolivia **Cloud (dancer)** [4 more candidates]	2 1 1 …
sebastien foucan	**Sébastien Foucan**	2
rize	Rize Rize (band) Rize Province **Rize (film)** [5 more candidates]	113 30 16 14 …

METHODOLOGY · CANDIDATE EVALUATION OVERVIEW

- Each candidate $r_{i,j}$ of m_i is assigned **two** scores:
  - **Initial Score**, $p(r_{i,j})$: Based on the local features of the textual mention and the intrinsic popularity of the candidate.
  - **Propagation Score**, $s(r_{i,j})$: Based on the collective, topical interaction of $r_{i,j}$ with the candidate mapping entities of the other mentions m_ℓ, $\ell \neq i$.
- A **final selection** is made based on the **most discriminant combination** of initial and propagation score.

INITIAL SCORE · PRIOR PROBABILITY

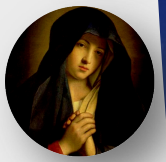
- Using the *NED* dictionary, compute for each candidate $r_{i,j}$

$$\Pr(r_{i,j}) = \frac{\text{count}(r_{i,j})}{\sum_{c=1}^{|R_i|} \text{count}(r_{i,c})}$$

- Captures the notion of entity popularity for a given mention m_i.
- We chose to limit the number of candidates per mention, experimentally, to **120**.

0.94

0.03



0.01

INITIAL SCORE · CONTEXT SIMILARITY[1]

- Tokenize the knowledge base entity documents and input text.
  - We wrote our version of the **CoNLL 2003 NER** tagging task tokenizer to process all of the Wikipedia articles and the local contexts.
- We extract a window of **50 tokens** around each occurrence m_i to create the **local contexts** of the surface forms M in the input text.
- Using pre-trained **fastText** word embeddings, calculate **context embeddings** after the proposed algorithm in [3] (also known as *Smooth Inverse Frequency*).

INITIAL SCORE · CONTEXT SIMILARITY^2

Context Embedding Algorithm

Input: Word embeddings $\{\mathbf{v}_w | w \in \mathcal{V}\}$, a set of contexts $\mathcal{C} = \{c_{m_i} \mid m_i \in M\} \cup \{c_{r_{i,j}} \mid r_{i,j} \in R_i\}$, $i = 1, \ldots, |M|$, parameter a, and estimated probabilities $\{p(w) | w \in \mathcal{V}\}$ of the words

Output: Context embeddings $\{\mathbf{v}_c | c \in \mathcal{C}\}$

for all context $c \in \mathcal{C}$ **do**

$$\mathbf{v}_c \leftarrow \frac{1}{|c|} \sum_{w \in c} \left(\frac{a}{a + p(w)} \mathbf{v}_w \right)$$

end for

Form a matrix X whose columns are $\{\mathbf{v}_c | c \in \mathcal{C}\}$, and let $\mathbf{u}$ be its first singular vector

for all context $c \in \mathcal{C}$ **do**

$$\mathbf{v}_c \leftarrow \mathbf{v}_c - \mathbf{u}\mathbf{u}^T \mathbf{v}_c$$

end for

Then use **cosine similarity** to calculate $\mathrm{Sim}(r_{i,j})$ of the candidate context embedding with respect to its surface form m_i's context embedding

METHODOLOGY · INITIAL SCORE

Convex combination of prior probability and context similarity:

$$p(r_{i,j}) = \alpha \Pr(r_{i,j}) + \beta \mathrm{Sim}(r_{i,j})$$

where $\alpha + \beta = 1$. Experimentally, $\alpha = 0.4$ and $\beta = 0.6$.

Madonna	Johan Renck	Daniel Campos	LaChapelle	Rize	Sebastien Foucan
Madonna (entertainer)	***Bo Johan Renck***	***Cloud (dancer)***	***David LaChapelle***	Rize	***Sébastien Foucan***
✓	✓	✓	✓	✗	✓

METHODOLOGY · PROPAGATION SCORE[1]

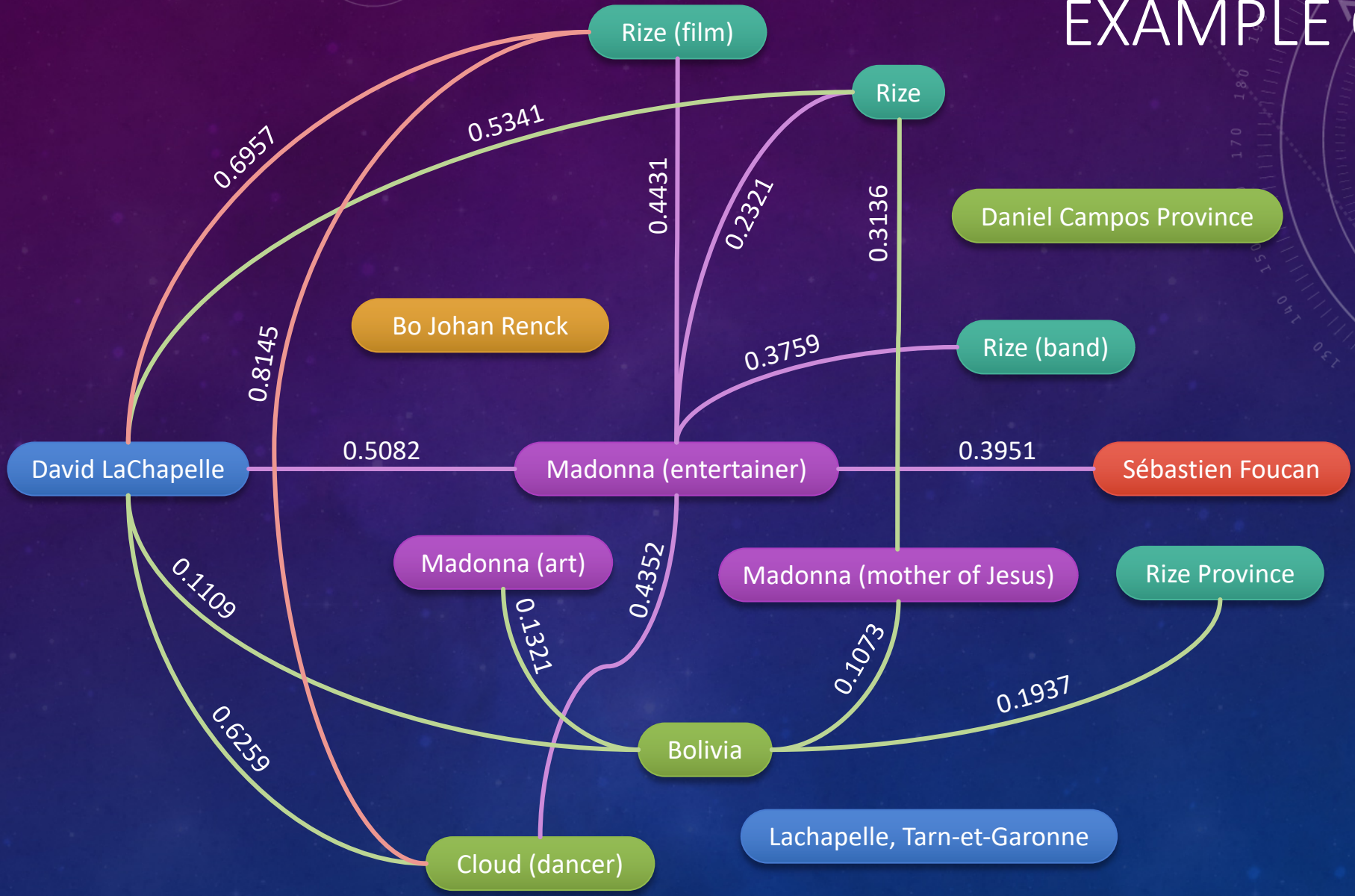
- Construct an undirected graph $G(V, W)$ whose edges W connect candidate mapping entities R_i with all other R_k, for all $i \neq k$, with edge strength given by the **Wikipedia Link-based Measure**:

$$TR(u_1, u_2) = 1 - \frac{\log(\max(|U_1|, |U_2|)) - \log(|U_1| \cap |U_2|)}{\log(|WP|) - \log(\min(|U_1|, |U_2|))}$$

where U_ℓ is the set of Wikipedia articles that link to u_ℓ, and WP is the set of all entity pages.

- Each $v_k \in V$ is one candidate mapping entity and has an initial score p_k.
- Our goal is to compute a "final" node (e.g. candidate mapping) score by propagating p_k through the edges leaving v_k.

EXAMPLE GRAPH



METHODOLOGY · PROPAGATION SCORE^2

- Normalize all nodes' initial score:

$$np_k = \frac{p_k}{\sum_{c=1}^{|V|} p_c}$$

- Let $\mathbf{p} = \left[np_1, \ldots, np_{|V|}\right]^T$ be a vector of normalized initial scores, and $B \in \mathbb{R}^{|V|\times|V|}$ be the matrix of edge strengths with normalized columns, where $b_{ij} \in B$ is the propagation strength from node v_j to node v_i.
- The final score of the nodes in V (and thus of candidate mapping entities) is computed using a **Personalized PageRank Algorithm** as follows:

$$\mathbf{s} = \lambda\mathbf{p} + (1-\lambda)B\mathbf{s}$$

where $\lambda = 0.4$ in our experiments.

METHODOLOGY · MAXIMAL DISCRIMINANT SELECTION

- After computing $\mathbf{s}$, we re-rank candidate mapping entities following the procedure in [1].
- Let $\mathbf{l}_s(R_i) = \mathbf{p}(R_i) + \mathbf{s}(R_i)$ and $\mathbf{l}_m(R_i) = \mathbf{p}(R_i) * \mathbf{s}(R_i)$ be two combination schemes of the initial and propagation scores of candidate mapping entities for m_i.

Re-Ranking Algorithm

Input: Two lists: L_s and L_m of candidates R_i for m_i,
where L_s is ranked using $\mathbf{l}_s$, and L_m is ranked using $\mathbf{l}_m$
Output: Entity mapping e_i for m_i
Sort L_s and L_m in descending order
    $\delta_s \leftarrow L_s[0] - L_s[1]$
    $\delta_m \leftarrow L_m[0] - L_m[1]$
if $\delta_s > \delta_m$ **then**
    return $R_i(L_s[0])$
else
    return $R_i(L_m[0])$

Madonna
Madonna (entertainer)

LaChapelle
David LaChapelle

Sebastien Foucan
Sébastien Foucan

Johan Renck
Bo Johan Renck

Daniel Campos
Cloud (dancer)

Rize
Rize (film)

IMPLEMENTATION DETAILS

WIKIPEDIA
The Free Encyclopedia

- We developed our NED system with **Wikipedia**[1] as Knowledge Base. Our framework is written in **Python** and uses **MongoDB** as underlying DB engine.
- Downloaded, parsed, and tokenized the **November 6, 2014 English Wikipedia XML** dump file.
  - Collected 4’589,969 entities.
  - Extracted 2’519,370 unique tokens with a grand total of 2,071’472,445 uses for SIF calculations.
  - Gathered 11’964,048 distinct surface forms.
  - DB size reached 27.11GB, split into 5 MongoDB collections.
- Use pretrained **300-dimensional** word embeddings from fastText[2].

mongo DB

[1]https://archive.org/download/enwiki-20141106
[2]https://fasttext.cc/docs/en/pretrained-vectors.html

RESULTS

- We used the **CoNLL 2003**[3] shared task for NER dataset to evaluate and tune the parameters of our system.
  - The dataset consists of **1387** Reuters newswire documents from fall 1995, manually annotated for NER and NED.
  - We only considered accuracy tests for surface forms **with at least 1 candidate mapping entity**.
  - There is a total of **18407** unique surface forms from which **399** mentions map to NIL.

Test	Correctly mapped	Macro accuracy	Micro accuracy
Full (initial score + propagation score + re-ranking)	13,926	77.3	**80.28**
No re-ranking	13,541	75.2	78.02
No propagation	13,248	73.6	75.49

[3]https://www.clips.uantwerpen.be/conll2003/ner/

CONCLUSIONS

- Learning outcomes:
  - Use word embeddings to compute context/document embeddings.
  - Use personalized page rank to find best candidate mappings.
  - Carry out NED by combining traditional techniques (e.g. prior probabilities, parsing/tokenization of a large corpus) with latest techniques like NoSQL and multiprocessing.
- Challenges and future work:
  - Investigate how to handle the case of same candidate mapping entity for different entity mentions.
  - Increase accuracy by adding more candidate mapping entity features.
  - Increase accuracy by re-engineering the process of candidate extraction to counteract noise and sparsity in Wikilinks.
    - More robust Wikipedia analyzer.
    - Validate whether document embeddings are “the” replacement for TF-IDF-based context similarity measurement.
  - How to pose the problem of parameter tuning as a machine learning optimization process.
  - How to solve the NED completely with machine learning.