Advancing NLP via a distributed-messaging approach

Ilaria Bordin, Andrea Ferretti, Marco Firrincieli, Francesco Gullo, Marcello Paris, Stefano Pascolutti, and Gianluca Sabena

UniCredit, R&D Dept., Italy

{ilaria.bordino, andrea.ferretti2, marco.firrincieli, francesco.gullo, marcello.paris, stefano.pascolutti, gianluca.sabena}@unicredit.eu

2016 IEEE International Conference on Big Data
December 5-8, 2016
Washington D.C., USA
Natural Language Processing (NLP)

“Set of techniques for automated generation, manipulation and analysis of human (natural) languages”

Major tasks:

- Language modeling
- Part-of-speech (POS) tagging
- Entity recognition and disambiguation
- Sentiment analysis
- Word sense disambiguation
Motivations:

- **Architectural limitations**: Existing solutions are stand-alone components focusing on specific micro-tasks, nor really suitable for distributed environments and large-scale data processing.

- **Algorithmic limitations**: Existing entity recognition and disambiguation (core NLP task) methods not really amenable to be deployed in a real-world industrial context (weaknesses in terms of both efficiency and result interpretability).
Contributions:

- We design Hermes, a novel NLP tool that overcomes the aforementioned state-of-the-art limitations

  **Architectural contribution**: Efficient and extendable architecture whose modules interact via message passing

    - Three major requirements satisfied:
      - capability of large-scale processing, completeness, versatility

  **Algorithmic contribution**: Novel solutions to entity recognition and disambiguation aiming at both efficiency and result interpretability
Architecture
Message queues

- Queues, producers, consumers
- Implementation details: Scala, Apache Kafka, JSON
Producers

- Retrieve the text sources to be analyzed, and feed them into the system
- Four different source types are currently supported:
  1. Twitter
  2. News articles
  3. Documents
  4. Mail messages
- Producers perform minimal processing and push on the text queue
Cleaner

- Consumes raw texts pushed on the text queue
- Performs text extraction
- Pushes extracted text onto the *clean-text* queue
- Implementation details: Goose for text extraction, Tika for content extraction and language recognition
NLP Module

- Handles sentence splitting, tokenization, HTML/Creole parsing, entity linking, topic detection, clustering of related news, sentiment analysis

- *Client/Server Design*: The client news on the clean-news queue, asks for NLP annotations to the service, and places the result on the tagged-news queue
Persister and Indexer

- Index service (ElasticSearch)
- Key-value store (HBase)
- Two long-running (Akka) applications listen to the clean-text and tagged-text queues, and respectively index and persist raw and decorated news
A single-page client (written in Coffee-Script using Facebook React) interacts with a Play application.

The client home page shows annotated news ranked by a relevance function that combines various metrics but users can also search.

The Play application retrieves news from the index and enriches them with content from the key-value store.
Algorithms
NLP: dealing with (named) entities

**Entity:** concept of interest in a text (e.g., a person, a place, a company)

**Entity Recognition and Disambiguation (ERD):**

- **Entity Recognition (ER):**
  identification of (candidate) entities in a plain text (i.e., which parts of the text to be linked)

- **Entity Disambiguation (ED), aka Entity Linking (EL):**
  resolving (i.e., “linking”) named entity mentions to entries in a structured knowledge base

*Non-uniform terminology: in some cases EL ≡ ERD*
Entity linking: scenario

M. J. Jordan (basketball)
M. B. Jordan (actor)
Basketball
National Basketball Association
National Bank Act
National Boxing Association
Earvin “Magic” Johnson Jr.
Wikipedia hyperlink graph

Mentions
Entities

Hermes: A distributed-messaging tool for NLP
Entity linking: voting approach

**Wikify!** [Mihalcea and Csomai, CIKM’07]
**Tagme** [Ferragina and Scaiella, CIKM’10]
**Wat** [Piccinno and Ferragina, ERD’14]

**Main idea**

Compute a score for each candidate mention-entity linking $a \mapsto e$ (based on the other possible mention-entity linkings $b \mapsto e'$ derived from the input text), and link each mention $a$ to the entity $e^*$ that maximizes that score, i.e., $e^* = \arg \max_e \text{score}(a \mapsto e)$. 
Voting-based entity linking: critical steps

- \( rel(e_1, e_2) = 1 - \frac{\max\{\log |\text{in}(e_1)|, \log |\text{in}(e_2)|\} - \log |\text{in}(e_1) \cap \text{in}(e_2)|}{|W| - \min\{\log |\text{in}(e_1)|, \log |\text{in}(e_2)|\}} \)

\( \Rightarrow \mathcal{O}(\min\{\text{deg}(e_1), \text{deg}(e_2)\}) \)

- \( \text{score}(a \mapsto e) = \sum_{b \in \mathcal{M}_T \setminus \{a\}} \text{vote}(a \mapsto e | b) = \frac{1}{|E(b)|} \sum_{b \in \mathcal{M}_T \setminus \{a\}, \ e' \in E(b)} \text{rel}(e, e') \Pr(e' | b) \)

for all possible \( a \mapsto e \)

\( \Rightarrow \mathcal{O}(N^2) \) (\( N = \sum_{m \in \mathcal{M}_T} |E(m)| \))
**Problem:** given two entities $e_1$ and $e_2$, and their corresponding neighbor sets $\mathcal{N}_1$ and $\mathcal{N}_2$ (with $|\mathcal{N}_1| = \text{deg}(e_1)$, $|\mathcal{N}_1| = \text{deg}(e_2)$), quickly estimate $|\mathcal{N}_1 \cap \mathcal{N}_2|$.

**Offline** ($n$: #entities, $m$: #edges in the entity-interaction graph (e.g., Wikipedia)):

- Choose $K$ hash functions $h^{(1)}, \ldots, h^{(K)} \rightarrow \mathcal{O}(Kn)$
  - basically, if our universe $U = \{1, \ldots, n\}$ corresponds to the id of the $n$ entities in our dataset, each $h^{(i)}$ is a random permutation of $U$

- Compute **min-hash signature** of each entity $e$ as a $K$-dimensional real-valued vector $\vec{v}_e = [h_{\min}^{(1)}(\mathcal{N}(e)), \ldots, h_{\min}^{(K)}(\mathcal{N}(e))] \rightarrow \mathcal{O}(K \sum e \text{deg}(e)) = \mathcal{O}(Km)$

**Online:**

- Estimate $J(\mathcal{N}(e_1), \mathcal{N}(e_2))$ as $\frac{1}{K} \sum_{i=1}^{K} 1[\vec{v}_{e_1}(i) = \vec{v}_{e_2}(i)]$
- Estimate $|\mathcal{N}(e_1) \cap \mathcal{N}(e_2)|$ as $\frac{J}{1+J}(|\mathcal{N}(e_1)| + |\mathcal{N}(e_2)|)$
- $\rightarrow \mathcal{O}(K)$ (rather than $\mathcal{O}(\min\{\text{deg}(e_1), \text{deg}(e_2)\})$)
LSH to speed-up voting-based EL

Offline:
- Compute LSH buckets $lsh(e) = [b_1(e), \ldots, b_L(e)]$ for each entity $e$, where $b_i(e) = lsh(i, \text{minhash}(e)) \rightarrow [\mathcal{O}(Kn) / L \approx \mathcal{O}(K)]$ (cf. [O(Km)] for MinHash).

Online (given an input text $T$):
- Retrieve LSH buckets for all entities in $T$
- Compute inverted index: for each bucket $b$, $\text{entities}(b) = \{e \mid b(e) \in lsh(e)\}$
- Approximate $\text{score}(a \mapsto e) = \frac{1}{|E(b)|} \sum_{b \in \mathcal{M}_T \setminus \{a\}, \ rel(e, e') \Pr(e' \mid b)}$ as $\frac{1}{|E(b)|} \sum_{e' \in \text{buckets}(e)} \rel(e, e') \Pr(e' \mid b)$

Instead of $\mathcal{O}(N^2)$ comparisons, only need comparisons between entities in the same bucket.
### Experiments

<table>
<thead>
<tr>
<th>Wikipedia</th>
<th>Exact</th>
<th>LSH</th>
<th>LSH-MinHash</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.825</td>
<td>0.808</td>
<td>0.806</td>
</tr>
<tr>
<td>R</td>
<td>0.629</td>
<td>0.638</td>
<td>0.629</td>
</tr>
<tr>
<td>FM</td>
<td>0.697</td>
<td>0.692</td>
<td>0.685</td>
</tr>
<tr>
<td>Tsetup (ms)</td>
<td>80050</td>
<td>115844</td>
<td>45111</td>
</tr>
<tr>
<td>Tdis (ms)</td>
<td>758</td>
<td>153</td>
<td>32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RSS</th>
<th>Exact</th>
<th>LSH</th>
<th>LSH-MinHash</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.67</td>
<td>0.65</td>
<td>0.65</td>
</tr>
<tr>
<td>R</td>
<td>0.66</td>
<td>0.64</td>
<td>0.64</td>
</tr>
<tr>
<td>FM</td>
<td>0.663</td>
<td>0.643</td>
<td>0.643</td>
</tr>
<tr>
<td>Tsetup (ms)</td>
<td>37697</td>
<td>50836</td>
<td>27950</td>
</tr>
<tr>
<td>Tdis (ms)</td>
<td>67</td>
<td>49</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reuters</th>
<th>Exact</th>
<th>LSH</th>
<th>LSH-MinHash</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.676</td>
<td>0.603</td>
<td>0.604</td>
</tr>
<tr>
<td>R</td>
<td>0.524</td>
<td>0.479</td>
<td>0.478</td>
</tr>
<tr>
<td>FM</td>
<td>0.579</td>
<td>0.525</td>
<td>0.525</td>
</tr>
<tr>
<td>Tsetup (ms)</td>
<td>198734</td>
<td>232641</td>
<td>66847</td>
</tr>
<tr>
<td>Tdis (ms)</td>
<td>3106</td>
<td>370</td>
<td>34</td>
</tr>
</tbody>
</table>
Check out our tool at hermes.rnd.unicredit.it:9603
(Email me (francesco.gullo@unicredit.eu) to get access credentials)

Thanks!