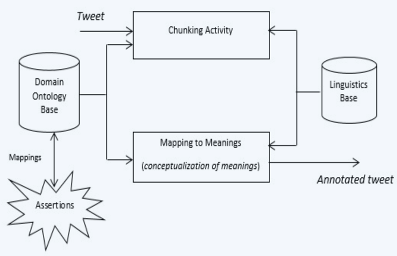
UNIVERSITÀ DEGLI STUDI DI PALERMO
DIPARTIMENTO DELL'INNOVAZIONE INDUSTRIALE E DIGITALE
Ingegneria Chimica | Gestionale | Informatica | Meccanica
diid

AI*IA 2017
16° International Conference of the
Italian Association for Artificial Intelligence
Bari, November 14-17, 2017

Named Entity Recognition and Linking in Tweets based on Linguistic Similarity

Arianna Pipitone, Giuseppe Tirone, and Roberto Pirrone
{arianna.pipitone, giuseppe.tirone, roberto.pirrone}@unipa.it

Named Entity rEcognition and Linking (NEEL) is a sub-task of information extraction that aims at locating and classifying each named entity mention in a tweet into the classes of a knowledge base, such as DBPedia.
According to [1], NEEL consists of ***mention detection*** related to the identification of the entity mention in a tweet, and ***candidate selection*** related to the identification of the link in DBPedia that defines such an entity.
The figure shows the proposed architecture.



The proposed system ranked second when compared to the outcomes of the #Micropost2016 workshop NEEL Challenge [1].

Rank	Approach	Team Name	F_1^{mc}	F_1^{stmm}	F_1^{slm}	*score*
1	sup	kea	0.641	0.473	0.501	0.5486
2	**unsup**	**our**	**0.616**	**0.515**	**0.406**	**0.5227**
3	unsup	insight-centre @ nuig	0.621	0.246	0.202	0.3828
4	sup	mit lincoln laboratory	0.366	0.319	0.396	0.3609
5	sup	ju team	0.467	0.312	0.248	0.3548
6	sup	unimib	0.203	0.267	0.162	0.3353
*	*sup*	*adel*	*0.69*	*0.61*	*0.536*	*0.6198*

Mention Detection

The *Chunking Activity* module outputs **all the possible words in the tweet to be analyzed for entity mention detection.**
The basic considerations are:

- Given a tweet *t*, its components can be classified in two categories based on <u>*the linguistic properties of the inherent chunks*</u>, that are:

  1. $M = \{m_i \mid m_i \text{ is a micropost of } t\}$, which contains both the main post that generates a discussion, and all the posts in the thread; chunks can be identified by blank spaces between words;
  2. $H = \{h_i \mid h_i \text{ is a hashtag or a tag of } t\}$, which contains the hashtags and the tags in *t;* chunking is not trivial, because no typical separation characters are used.

- <u>*Informal language can influence linking*</u>: the chunks devised so far must be rewritten using words already owned by the system. For this purpose, automatic correction[1] based on the WordNet source is applied to the identified chunks.

Formally, let be:

- *tok(s)* the function that returns the list of tokens L_t split using blank spaces for the string s;
- *a_star(s)* the function that returns the list of chunks L_c for the string *s* based on the $A*$ strategy reported in [2];
- *icbl(k)* the function that returns the list L_w of the words that are syntactically similar ($\cong$) to the token *k,* using the automatic correction[1].

The Chunking Activity module implements the functions:

$$c_a : H \cup M \rightarrow L_c$$

$$c_a(s) = \begin{cases} a\_star(s) & s \in H \\ a\_star(tok(s)) & s \in M \end{cases}$$

$$icbl : L_c \rightarrow L_w, \quad icbl(k) = \quad \{w_i \mid w_i \cong k\}$$

whose output is the set $C = H \cup M \cup L_w$.

It is well acknowledged [1] that *an entity in a tweet can be only a proper noun* (NP or NPS), and a POS tagger[2] is applied to the words in C for identifying the possible candidates to be a mention. The process ends with the definition of the set MD that will contain all candidate mentions:

$$MD = \{m_i \mid m_i = \{c_j, c_{j+1}, ..., c_{j+n_i}\} \subset C,\ pos(m_i) \in \{NP, NPS\}\}$$

being the n_i value the extent of the *i-th* mention.

Candidate Selection

The *Mapping to Meanings* module from QuASIt [3] is adapted for candidate selection; the $a_{c_{neel}}$ function returns, for each mention in MD, the best matching entities in DBPedia:

$$a_{c_{neel}}(m_i) = \{\, cl_k \mid stem(m_i) = stem(i_j) \vee sim(concat(m_i), i_j) > \tau,\ i_j = map(cl_k),\ cl_k \in C_o\}$$

where:

- *stem(w)* returns the stem of the word *w*;
- *sim(w* $sim(w_1,w_2)$ returns the distance between two words by combining their Jaro-Winkler[3] and Levenstein[3] distances:

  $$sim(w_1,w_2) = 0.5*jaro(w_1,w_2)+ 0.5*lev(w_1,w_2)$$

  The τ value was experimentally fixed to 0.7 as better threshold for *sim(,);*
- *concat(m)* returns the chunks concatenation in a mention *m*;
- *map(c)* returns set $I = \{i_j\}$ containing the instances in DBPedia whose stem of their class label is similar or equal to a mention stem in MD.
- C_o is the set of class names in DBPedia.

The set $\cup_{MD}\, a_{c_{neel}}(m_i) \cup I$ is the *assertion graph* of the tweet in DBPedia that realizes our NEEL task.

References
[1] Rizzo, G., van Erp, M., Plu, J., Troncy, R.: Making sense of microposts (#microposts2016) named entity recognition and linking (NEEL) challenge. In: Dadzie et al.
[2] Pipitone, A., Campisi, M.C., Pirrone, R.: An A* based semantic tokenizer for increasing the performance of semantic applications. In: 2013 IEEE Seventh International Conference on Semantic Computing, Irvine, CA, USA, September 16-18,
2013. pp. 393{394. IEEE Computer Society (2013)
[3] Pipitone, A., Tirone, G., Pirrone, R.: Quasit: A cognitive inspired approach to question answering for the italian language. In: AI*IA 2016 Advances in Artificial Intelligence, pp. 464{476. Springer International Publishing
[1] https://icbld.com
[2] https://courses.washington.edu/hypertxt/csar-v02/penntable/
[3] https://www.joyofdata.de/blog/comparison-of-string-distance-algorithms/