Multilingual Information Extraction with POLYGLOTIE

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yonas Kbrom, Yunyao Li, Huaiyu Zhu
IBM Research - Almaden
{akbika,chiti,mdanile,yktbrom,yunyaoli,huaiyu}@us.ibm.com

Abstract

We present POLYGLOTIE, a web-based tool for developing extractors that perform Information Extraction (IE) over multilingual data. Our tool has two core features: First, it allows users to develop extractors against a *unified abstraction* that is shared across a large set of natural languages. This means that an extractor needs only be created once for one language, but will then run on multilingual data without any additional effort or language-specific knowledge on part of the user. Second, it embeds this abstraction as a set of views within a declarative IE system, allowing users to quickly create extractors using a mature IE query language. We present POLYGLOTIE as a hands-on demo in which users can experiment with creating extractors, execute them on multilingual text and inspect extraction results. Using the UI, we discuss the challenges and potential of using unified, crosslingual semantic abstractions as basis for downstream applications. We demonstrate multilingual IE for 9 languages from 4 different language groups: English, German, French, Spanish, Japanese, Chinese, Arabic, Russian and Hindi.

1 Introduction

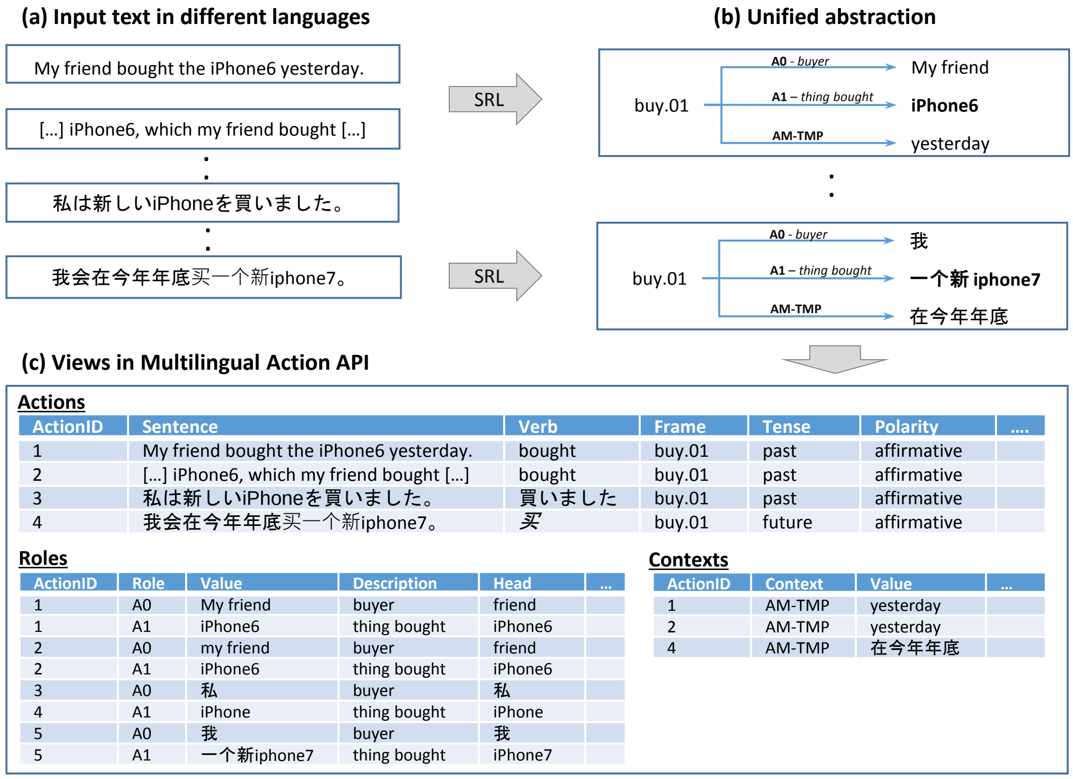
Information Extraction (IE) is the task of automatically extracting structured information from text (Sarawagi, 2008). Current IE approaches mostly focus on monolingual data and use language-specific feature sets to create extractors (Mintz et al., 2009; Surdeanu and Ji, 2014; Rocktäschel et al., 2015). A downside of such approaches is that extractors need to be separately created for each new language of interest, potentially blowing up costs.

With this demo, we present POLYGLOTIE, a web-based tool that allows users to create extractors over a *unified, crosslingual abstraction* of shallow semantics. The core advantage of our approach is that extractors need only be created once for one language against this abstraction, but can then automatically extract information from multilingual text.

We base our approach on previous work in multilingual semantic parsing (Akbik et al., 2015; Akbik and Li, 2016a; Akbik and Li, 2016b). In this research, we created a semantic role labeler (SRL) capable of predicting shallow semantic frame and role labels from the English Proposition Bank (Palmer et al., 2005) for 9 languages from 4 different language groups. We propose to utilize these semantic labels as the shared feature set against which users develop extractors. This, we argue, has two advantages: First, semantic role labels have human readable, shallow semantic descriptions (such as *buyer*, *thing bought*, and *price paid*) allowing users even without a background in linguistics to develop extractors. Second, since the same English labels are detected across all languages, users need not be language experts in a target language to create extractors. For instance, an English speaker might use this abstraction to create extractors for Chinese or Japanese.

The purpose of the demo is twofold: a) We demonstrate how extractors can be formulated against a shared abstraction based on frame semantics, and illustrate how they extract information from multilingual text. b) We illustrate the challenges and potential of using a frame-semantic abstraction for crosslingual applications.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 268–272, Osaka, Japan, December 11-17 2016.


(a) Input text in different languages
My friend bought the iPhone6 yesterday.
[...] iPhone6, which my friend bought [...]
私は新しいiPhoneを買いました。
我会在今年年底买一个新iphone7。
SRL
SRL
(b) Unified abstraction
buy.01
A0 - buyer
A1 – thing bought
AM-TMP
My friend
iPhone6
yesterday
buy.01
A0 - buyer
A1 – thing bought
AM-TMP
我
一个新 iphone7
在今年年底
(c) Views in Multilingual Action API

Actions

ActionID	Sentence	Verb	Frame	Tense	Polarity	....
1	My friend bought the iPhone6 yesterday.	bought	buy.01	past	affirmative	
2	[...] iPhone6, which my friend bought [...]	bought	buy.01	past	affirmative	
3	私は新しいiPhoneを買いました。	買いました	buy.01	past	affirmative	
4	我会在今年年底买一个新iphone7。	买	buy.01	future	affirmative	

Roles

ActionID	Role	Value	Description	Head	...
1	A0	My friend	buyer	friend	
1	A1	iPhone6	thing bought	iPhone6	
2	A0	my friend	buyer	friend	
2	A1	iPhone6	thing bought	iPhone6	
3	A0	私	buyer	私	
4	A1	iPhone	thing bought	iPhone	
5	A0	我	buyer	我	
5	A1	一个新iphone7	thing bought	iPhone7	

Contexts

ActionID	Context	Value	...
1	AM-TMP	yesterday	
2	AM-TMP	yesterday	
4	AM-TMP	在今年年底	

Figure 1: A multilingual text collection consisting of English, Japanese and Chinese sentences **(a)** parsed into the unified shallow semantic abstraction given by PropBank labels **(b)**. The abstraction is exposed through three views in POLYGLOTIE **(c)**.

2 A Unified Crosslingual Abstraction

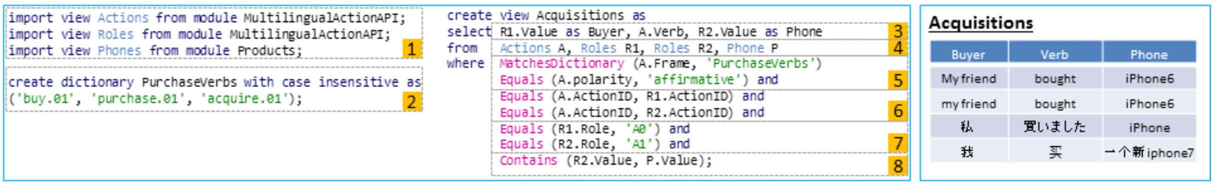
Frame semantics as language-independent abstraction. We utilize POLYGLOT (Akbik and Li, 2016a), a semantic role labeler that predicts English Proposition Bank frame and role labels for sentences in one of 9 different languages. The SRL is trained with target language data that was automatically labeled with English PropBank labels using an annotation projection approach (Akbik et al., 2015; Akbik and Li, 2016b).

Refer to Figure 1 for an illustration of this process. Input text in three different languages **(a)** is parsed into a frame-semantic representation with English labels. The representation is illustrated in Figure 1 **(b)** for two of the four input sentences, an English and a Chinese sentence. In all sentences, the BUY.01 frame is recognized, together with the roles *buyer* and *thing bought* and a temporal context. Crucially, after parsing into the unified abstraction, no language specific shallow semantic features remain.

Exposing Views. We execute POLYGLOT over the multilingual corpus and expose the frame-semantic representation of all sentences through a simple, programmable API in three views. See Figure 1 **(c)** for an illustration of these views. Each view carries a number of attributes: An ACTIONS view that exposes information on frame-evoking verbs, including the frame (BUY.01), the tense (past, present, future) and the polarity (affirmative or negative). A ROLES view that exposes the primary arguments of verbs (PropBank roles A0 through A4), including information on syntactic argument structure. And a CONTEXTS view that exposes information about adjuncts of verbs such as temporal, location and manner contexts, corresponding to optional roles in PropBank.

3 Declarative Information Extraction Against the Unified Abstraction

We create extractors in a declarative fashion against these views. In declarative IE, extractors are fundamentally SQL-like queries against views that create other views that are either output as extraction results or embedded in other extractors (Chiticariu et al., 2010). This approach has the advantages of providing



```
import view Actions from module MultilingualActionAPI;
import view Roles from module MultilingualActionAPI;
import view Phones from module Products;

create dictionary PurchaseVerbs with case insensitive as
('buy.01', 'purchase.01', 'acquire.01');
```

```
create view Acquisitions as
select R1.Value as Buyer, A.Verb, R2.Value as Phone
from   Actions A, Roles R1, Roles R2, Phone P
where  MatchesDictionary (A.Frame, 'PurchaseVerbs')
       Equals (A.polarity, 'affirmative') and
       Equals (A.ActionID, R1.ActionID) and
       Equals (A.ActionID, R2.ActionID) and
       Equals (R1.Role, 'A0') and
       Equals (R2.Role, 'A1') and
       Contains (R2.Value, P.Value);
```

Acquisitions

Buyer	Verb	Phone
My friend	bought	iPhone6
my friend	bought	iPhone6
私	買いました	iPhone
我	买	一个新 iphone7

Figure 2: Multilingual extractor for smartphone acquisitions. AQL rule and extraction results.

a standard IE language and data model, and allowing for the creation of succinct and embeddable views, further simplifying the process of developing multilingual extractors.

For an example of a declarative extractor, refer to Figure 2. This extractor searches for instances of a relation between buyers and the smartphone they purchase. To illustrate how the extractor works, we give details on each block of lines, referred to by the numbered blocks (**1**)-(**8**) in the figure. We first import a series of *views* (previously defined extractors or NLP components) through the import statements in block (**1**): Those views are the ACTIONS and ROLES views as defined in section 2, as well as the PHONES view, a previously created NER for smartphones. We then define a *dictionary* of acquisition-evoking frames (**2**), such as BUY.01 and PURCHASE.01. We then define the extractor as a view called ACQUISITION (**4**), which we create using the previously imported views (ACTIONS, ROLES and PHONES), with several constraints: ACTIONS need to be part of the previously defined dictionary (since we are only interested in *buying* actions) and the *polarity* should be positive (to discard negated actions such as *will not buy*) (**5**). ACTIONS is joined with two copies of ROLES to retrieve two roles for each frame (**6**), which we require to be 'A0' and 'A1' respectively (**7**). A final constraint is that the latter role should contain a mention of a smartphone, which we add by matching it to the PHONES view (**8**). Finally, we define the extractor output as the buyer, verb and phone retrieved from the relevant attributes in the input views (**3**). For example output of this extractor, refer to the "Acquisitions" table in Figure 2 (right hand side).

Background on the query language. The example was created using only two statements that make use of multiple built-in constructs of the Annotation Query Language (AQL): dictionary matching (CREATE DICTIONARY and MATCHESDICTIONARY constructs), span operations (the EQUALS and CONTAINS built-in predicates), and relational operations such as selection, projection and join (the SELECT, FROM and WHERE clauses). AQL is part of the SystemT (Chiticariu et al., 2010) framework for expressing NLP algorithms with both rules and machine learning constructs. For further reading, please refer to the ACL Reference[1].

4 Multilingual IE Web Interface and Demo Scenarios

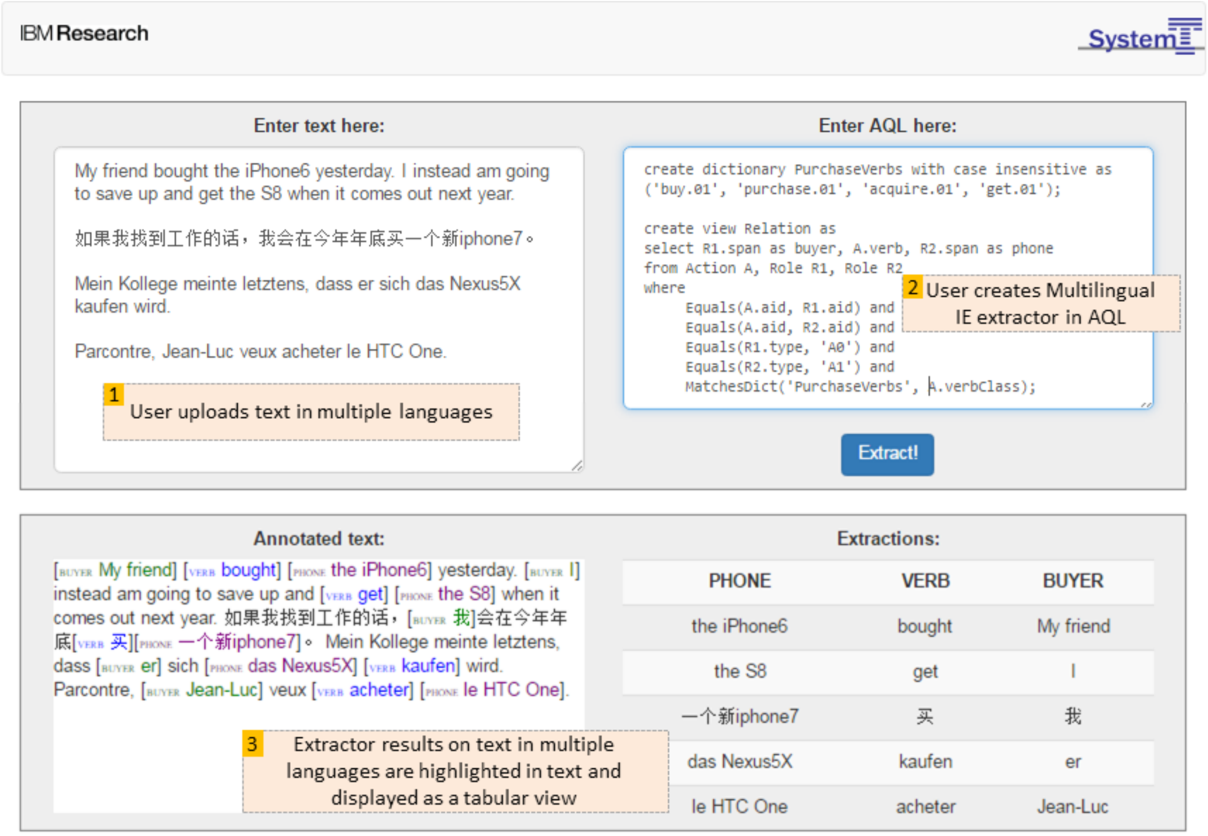
We present our web tool as a hands-on demo where users can create extractors and execute them on multilingual text. Refer to Figure 3 for an illustration of the web UI. In the top row, users can enter multilingual text **(1)** and create or modify an AQL rule that defines an output view **(2)**. Upon hitting the extract button, the rule is executed over the input and the results are visualized in two ways **(3)**: In the annotated text view, the extractions are annotated as labels in the input text. In the extractions view, the results are given in table format that shows the view as produced by the extractor.

We will use two demonstration scenarios. The first scenario involves the retail domain to identify purchase behavior similar to the smartphone acquision extractor discussed in Figure 1. The second scenario involves event extraction focused on the sports domain.

5 Discussion and Outlook

With this demonstration, users explore declarative IE rules over a unified, frame-semantic abstraction to create multilingual extractors. A point of discussion and current research is the coverage of semantic constructs required for IE applications. For instance, in its current form, the API is verb-centric, but

[1] The AQL Reference is available at: `http://ibm.co/2bNuweC`



Enter text here:

My friend bought the iPhone6 yesterday. I instead am going to save up and get the S8 when it comes out next year.

如果我找到工作的话，我会在今年年底买一个新iphone7。

Mein Kollege meinte letztens, dass er sich das Nexus5X kaufen wird.

Parcontre, Jean-Luc veux acheter le HTC One.

1 User uploads text in multiple languages

Enter AQL here:

```
create dictionary PurchaseVerbs with case insensitive as
('buy.01', 'purchase.01', 'acquire.01', 'get.01');

create view Relation as
select R1.span as buyer, A.verb, R2.span as phone
from Action A, Role R1, Role R2
where
      Equals(A.aid, R1.aid) and
      Equals(A.aid, R2.aid) and
      Equals(R1.type, 'A0') and
      Equals(R2.type, 'A1') and
      MatchesDict('PurchaseVerbs', A.verbClass);
```

2 User creates Multilingual IE extractor in AQL

Annotated text:

[BUYER My friend] [VERB bought] [PHONE the iPhone6] yesterday. [BUYER I] instead am going to save up and [VERB get] [PHONE the S8] when it comes out next year. 如果我找到工作的话，[BUYER 我]会在今年年底[VERB 买][PHONE 一个新iphone7]。 Mein Kollege meinte letztens, dass [BUYER er] sich [PHONE das Nexus5X] [VERB kaufen] wird. Parcontre, [BUYER Jean-Luc] veux [VERB acheter] [PHONE le HTC One].

3 Extractor results on text in multiple languages are highlighted in text and displayed as a tabular view

Extractions:

PHONE	VERB	BUYER
the iPhone6	bought	My friend
the S8	get	I
一个新iphone7	买	我
das Nexus5X	kaufen	er
le HTC One	acheter	Jean-Luc

Figure 3: Screenshot of the POLYGLOTIE web interface.

current work on frame-semantic abstractions (Banarescu et al., 2012; Bonial et al., 2014) focuses on other types of frame-evoking elements such as complex predicates. Furthermore, while our abstraction currently captures the semantic roles of constituents, lexical values often diverge between languages (the city of Milan for instance is called Milano in Italian and Mailand in German). Accordingly, we will focus on broadening our multilingual parsing to entity-level concepts, similar to entity-level annotations in abstract meaning representations (Banarescu et al., 2012).

References

A. Akbik and Y. Li. 2016a. Polyglot: Multilingual semantic role labeling with unified labels. In *ACL*.

Alan Akbik and Yunyao Li. 2016b. Towards semi-automatic generation of proposition banks for low-resource languages. In *EMNLP*.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating high quality proposition banks for multilingual semantic role labeling. In *ACL*, pages 397–407.

L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. 2012. Abstract meaning representation (amr) 1.0 specification. In *EMNLP*.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D Hwang, and Martha Palmer. 2014. Propbank: Semantics of new predicate types. In *LREC*, pages 3013–3019.

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. SystemT: An algebraic approach to declarative information extraction. In *ACL*, pages 128–137.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL/AFNLP*, pages 1003–1011.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *HTC-NAACL*.

Sunita Sarawagi. 2008. Information extraction. *Foundations and trends in databases*, 1(3):261–377.

M. Surdeanu and H. Ji. 2014. Overview of english slot filling track at TAC KB population evaluation. In *TAC*.