Language ID for a Thousand Languages

Fei Xia, Carrie Lewis
Department of Linguistics
University of Washington
{fxia,westplc}@uw.edu

William D. Lewis
Microsoft Research
wilewis@microsoft.com

Abstract

ODIN, the Online Database of INterlinear text, is a resource built over language data harvested from linguistic documents (Lewis, 2006). It currently holds approximately 190,000 instances of Interlinear Glossed Text (IGT) from over 1100 languages, automatically extracted from nearly 3000 documents crawled from the Web. A crucial step in building ODIN is identifying the languages of extracted IGT, a challenging task due to the large number of languages and the lack of training data. We demonstrate that a coreference approach to the language ID task significantly outperforms existing algorithms as it provides an elegant solution to the unseen language problem. We also discuss several issues that make automated Language ID and the maintenance of ODIN very difficult.

1 Introduction

A large number of the world's languages have been documented by linguists; it is now increasingly common to post current research and data to the Web, often in the form of language snippets embedded in scholarly papers. A particularly common format used to present language data analysis relevant to a particular argument or investigation is *Interlinear Glossed Text* (IGT). An example is shown in (1).

(1) Taro-wa John-ga kasiko-i-to omotta
Taro-TOP John-NOM smart-Pres-Comp think-past
"Taro thought that John was smart." Goro (2003)

ODIN, *the Online Database of INterlinear text*, is a resource built from data harvested from scholarly documents (Lewis, 2006). It was built in three steps: (1) crawling the Web to retrieve documents that may contain IGT, (2) extracting IGT from the retrieved documents, and (3) identifying the language codes and names of the extracted IGTs. The identified IGTs are then extracted and stored in a database (the ODIN database), which can be easily searched with a GUI interface. The database currently holds nearly 190,000 IGT instances representing language data for over 1,100 languages, extracted from nearly 3000 documents crawled from the Web. Among the three steps, the language identification (ID) step is the most challenging and is the focus of this paper.

2 Language names and codes

When dealing with thousands of languages, it is not sufficient to use language names to identify languages, because the mapping between language data and language names is not one-to-one. Many languages have several alternative names in addition to the primary names. For instance, the language Alumu-Tesu has alternative names such as Alumu, Arum-Cesu, Arum-Chessu, and Arum-Tesu.[1] Conversely, many language names can refer to multiple languages. For instance, the language name *Hmong* can refer to more than two dozen closely related languages

[1] http://www.ethnologue.com/show_language.asp?code=aab

spoken by different groups of Hmong. Given this fact, we decide to use language codes to identify the language data in ODIN, where a language code is a 3-letter code that *uniquely* identifies a language.

There are three existing language tables developed by the linguistics community: (1) ISO 639-3 maintained by SIL International,[2] (2) the 15th edition of the Ethnologue,[3] and (3) the list of ancient and dead languages maintained by LinguistList.[4] We merged the three tables, and the results are shown in Table 1. Out of 44,071 unique language names in the merged language table, 2625 of them (5.95%) are ambiguous.

Table 1: Language tables used in this study

Language table	# of lang codes	# of lang (code, name) pairs
(1) ISO 639-3	7702	9312
(2) Ethnologue v15	7299	42789
(3) LinguistList table	231	232
Merged table	7816	47728

3 Language ID algorithms

The goal of the language ID step is to assign a language (name, code) pair to each IGT instance extracted from linguistic documents. There have been extensive studies on language ID of written text, and a review of previous research on this topic can be found in (Hughes et al., 2006). All the existing language ID algorithms require a collection of text for training, something on the order of a thousand or more characters, a requirement that cannot be met in our setting (and possibly other settings dealing with resource poor languages) because about half of the languages in ODIN never appear in the data set that we used to train language ID algorithms – we call this the *unseen language problem*. As a result, these algorithms perform poorly in this particular setting. For instance, Cavnar and Trenkle's N-gram-based algorithm achieves an accuracy as high as 99.8% when tested on newsgroup articles across eight languages (Cavnar and Trenkle, 1994). However, its accuracy drops to only 51.4% on the ODIN data set.[5]

Because the language name associated with an IGT instance almost always appears *somewhere* in the document, we designed an algorithm that treats the language ID task as a coreference resolution problem, where IGT instances and language names appearing in the document are *mentions* and the language codes corresponding to these language names are *entities*. The algorithm simply needs to link mentions to entities, allowing us to apply any good coreference resolution algorithms (e.g., (Ng, 2005; Luo, 2007)) and providing an elegant solution to the unseen language problem. This approach significantly outperforms existing algorithms, especially when there is very little training data. The detail of the algorithm and experimental results can be found in (Xia et al., 2009).

4 Issues with the language table

To ensure the high quality of the ODIN database, the system output of the language ID module was manually corrected. The manual correction process reveals several problems with the language table, as explained below.

4.1 Incomplete language table

About one third of errors made by our language ID system are due to inadequacies of the language table in Table 1: entries are missing and some are even incorrect. There are three types of errors in the table.

Missing language names: Sometimes the correct language name for an IGT instance does not even appear in the language table. There are two possible reasons for that. First, authors of the ODIN document refer to a known

[2] http://www.sil.org/iso639-3/download.asp

[3] http://www.ethnologue.com/codes/default.asp#using

[4] http://linguistlist.org/forms/langs/GetListOfAncientLgs.html

[5] This data set contains 15,239 IGTs from ODIN whose correct language codes are provided by humans. We used 90% of data for training and the remaining 10% for testing.

language with a new language name due to spelling variation or other naming variations (e.g., Aroplokep vs. Arop-Lukep, Banka vs. Bankagooma, Old Greek vs. Ancient Greek). Second, for several dozen language names, the linguistics community has not assigned language codes to them. Some examples are Medieval Spanish, Greenlandic Pidgin, Early High German, Middle Swedish, Old Bangali, Tugu Creole, and Taimyr Pidgin Russian.

Missing pairs: The language table can miss certain (language name, language code) pairs. For instance, the name *St. Louis* can refer to two unrelated dialects: one is a dialect of Wolof (*wol*), a language spoken in in Senegal; the other is a dialect of Caac (*msq*), a language of New Caledonia. The language table includes only the pair *(St. Louis, msq)*, but not *(St. Louis, wol)*.

Wrong pairs: In some rare cases, the language table assigns wrong language codes to certain language names. For example, Yucatec Maya, which invariably refers to the language spoken in Mexico, is given the language code of its Yucatecan relative Itza, spoken in Guatemala, since the Mexican variety is listed as Yucatan Maya.

To address these problems, during manual correction we maintained a new table that specified the errors in the original table and showed the correct entries. By now, we have completed all manual correction and the new table includes 720 new language names, 900 new language pairs and 18 pairs in the original table that are wrong. The corrected data and the revised language table will be used to re-train our language ID system.

4.2 Ambiguous language names

A language name is ambiguous if it can map to multiple language codes. The most common reason for this ambiguity is that linguists use a generic language name (e.g., *Quechua*) to refer to a particular language (e.g., *Cusco Quechua*, *Imbabura Quichua*, or 42 other closely related languages). ISO 639-3 provides a list of 58 such generic names (a.k.a *macrolanguages*) and the corresponding 429 individual language names.[6] Another source of ambiguity is that some language names are used to refer to unrelated languages by accident: e.g., Tiwa (Sino Tibetan) and Tiwa (Tanoan) are two different languages. Dani (Trans-New Guinea), which may refer to four related languages of the same name, is often mistakenly used to refer to Deni (Arauan), an unrelated language.

Choosing the correct language code for some ambiguous language names is not trivial at all, even for linguistic experts. For instance, in order to determine which individual language a macrolanguage refers to, we start with the list of all the individual languages for that macrolanguage, and narrow it down by looking for cues within the document (e.g., the region that the language is spoken, other languages or dialects appearing in the same document, author's or cited author's language of study, and the annotator's knowledge about the differences between individual languages). If this process does not pinpoint one individual language, we will use the IGT instance to search the Web for documents sharing the same or a similar example and use the cues in those documents. If there are still multiple individual languages left after all this work, we keep all the remaining languages with the IGT instance.

4.3 Constant changes to language tables

The three existing language tables mentioned in Section 2 are subject to periodic changes and updates, which makes it very difficult for ODIN to maintain a consistent and up-to-date language table. In some instances the changes can be catastrophic, such as when Ethnologue changed their codes in v15 in order to align them with ISO 639-2's standards. A number of language codes in v14 were retired and reassigned to other language names. Consequently, ODIN, which used v14, had to undergo a major remapping, which also had a cumulative effect on language mappings of related and unrelated languages. Such was the case with Serbo-Croatian and the unrelated language Sardinian, whose language codes in v14 were *src* and *srd* respectively. In v15, Serbo-Croatian (*src*) was retired, but the code *src* was reassigned to Sardinian (Logudorese) and its former code *srd* was retired. In addition, the language Serbo-Croatian no longer has a language code in v15 and is now represented by three language codes: Croatian (hrv), Serbian (srp), and Bosnian (bos). To make matters worse, two of these codes were formerly used for unrelated languages Saruga (srp) and Bosilewa (bos). While 1-to-1 and n-to-1 mappings between the old and new versions of language tables

[6] http://www.sil.org/iso639-3/macrolanguages.asp

can be easily handled, complicated mapping illustrated by this example requires manual reassignment of previously verified language-code pairs. This is particularly difficult in ODIN because it often requires consulting the original source documents for verification of the authors intent (what language code did the author intend?). Furthermore, the problem is not resolvable when the authors specifically refer to Serbo-Croatian in their text, which is often the case: many linguists consider Serbian and Croatian to be dialects of one language, not separate languages.

Ethnologue's change from v15 to v16 (ISO 639-2 to ISO 639-3) is far less drastic. However, language codes are still subject to retirement and some language names have been reassigned. For instance, the language code for Malay in v15 was *mly*, but the code has been retired in v16. Malay in v16 has been split into four language (name, code) pairs: Haji (hji), Malay (zlm), Paupan Malay (pmy), and Standard Malay (zsm). The same problem presents itself for resolving these language codes as existed with Serbo-Croatian.

5 Conclusion

To summarize, we have demonstrated that the coreference approach to identifying the language names/codes of IGT in linguistic documents produces much better results than existing language ID algorithms. While a large number of languages and small sample sizes present difficult problems in and of themselves, incomplete language tables, ambiguous language names, and constant changes to language tables make automated Lang ID and the maintenance of ODIN even more difficult. Since the data for more and more languages is finding its way into digital form, with an increasing amount of this data being posted to the Web, resolving the crucial issues laid out here will have significant positive effects on linguistics, NLP, and related disciplines. Although we do not explicitly say so here, altering the means by which resources are posted to the Web and other data stores, including language data embedded in scholarly documents, specifically by using standard language names and codes, could have significant effects on how resources like ODIN can be built, and how linguists and other language researchers can find and use these data.

Acknowledgements We are grateful to the support from the National Science Foundation grants BCS-0720670 and BCS-0748919. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Cavnar, W. B. and Trenkle, J. M. (1994). N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US.

Goro, T. (2003). Japanese disjunction and positive polarity.

Hughes, B., Baldwin, T., Bird, S., Nicholson, J., and MacKinlay, A. (2006). Reconsidering language identification for written language resources. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 485–488, Genoa, Italy.

Lewis, W. D. (2006). ODIN: A Model for Adapting and Enriching Legacy Infrastructure. In *Proc. of the e-Humanities Workshop, held in cooperation with e-Science 2006: 2nd IEEE International Conference on e-Science and Grid Computing*, Amsterdam.

Luo, X. (2007). Coreference or not: A twin model for coreference resolution. In *Proc. of the Conference on Human Language Technologies (HLT/NAACL 2007)*, pages 73–80, Rochester, New York.

Ng, V. (2005). Machine learning for coreference resolution: From local classification to global ranking. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 157–164, Ann Arbor, Michigan.

Xia, F., Lewis, W. D., and Poon, H. (2009). Language ID in the Context of Harvesting Language Data off the Web. In *Proceedings of The 12th Conference of the European Chapter of the Association of Computational Linguistics (EACL 2009)*, Athens, Greece.