

MLSN: A multi-lingual semantic network

Darren Cook
darren@dcook.org

Introduction And Motivation

In many applications in computing knowing how words relate to other words can help us do useful work with them. If a user of a search engine types in "sports car" we want to know that implies they may also be interested in "Ferrari". This can be used not just to improve search results but also to help us choose suitable advertising to display. This example requires an understanding of what we refer to as "culture words"(in contrast to dictionary words): brands, companies, movies, books, famous people and so on.

English is the most common language on the internet, and in the world of business, but Japanese is the most common language in the world's second largest economy, and Chinese is the most common language in the world's latest economic superpower. The need to understand the relationships between words in all languages is clear. But this does not require duplication of all the effort. A フェラーリ (Ferrari) is still a type of スポーツカー (sports car) in Japanese. Only the words have changed, not the relation.

MLSN (MLSN, 2006) is an open source project, started in late 2005, to create a semantic network that includes both dictionary words and culture words, and that is multilingual, covering all the world's major languages. The project is first and foremost a database of words and their relations. But it is also a web-based front-end for people to view and enhance that database. It is also a set of utilities for users to make their own local version enhanced with, for instance, their company's confidential data. The open source license freely allows commercial use in the hope that some percentage of business users will contribute back or sponsor enhancements.

Design Decisions and Data Structures

Princeton's WordNet (Fellbaum, 1998) was chosen as the starting point as it is already fairly complete, at least for English dictionary words, and has a liberal license allowing the use of the data in commercial applications. Our first design decision was to follow WordNet's structure as much as possible, allowing us to benefit from future versions, to more easily relate to other WordNet-related projects, and make it possible to feed our additions back.

But already we meet our first challenge in our desire to have a *multi-lingual* semantic network. WordNet is subjective. This is not a criticism, it is a natural consequence of trying to describe a living language, but it is a problem. As an example the Japanese language has nouns for various types of sushi, so in a Japanese semantic network we naturally want one entry for each. But though WordNet has an entry for sushi (13sushi0) it has no hyponyms. Another example is the Japanese word "girichoko". This is chocolate given as an obligation (on Valentines Day and White Day) rather than for romantic reasons. Such words are culture-specific but are needed in a multi-lingual semantic network.

Our solution is simple: we just add girichoko (13girichoko0) and some hyponyms of 13sushi0. When there is no common English word the romanized form of the Japanese is used in the English version, and this generally decides the unique identifier (the lemma in WordNet terminology).

Our second challenge follows on almost immediately. WordNet is primarily for contemporary American English. As an example take the luggage compartment at the back of a car. This has the code 06luggage_compartment0 and has "luggage compartment, automobile trunk, trunk" as synonyms. The British English word for this is "boot", which in WordNet is given as 06boot1, a hyponym of 06luggage_compartment0. From a language-neutral point of view "boot" is really just a synonym of 06luggage_compartment0. When we translate into Japanese we have to put "トランク" for both entries, and when we translate into German we have to put "Gepäckraum [m]" for both entries. Our current solution is a non-solution: we just accept the duplicates.

WordNet includes nouns, verbs, adverbs and adjectives but our next design decision was to only handle nouns. (Noun is used with the meaning of a collocation, in WordNet terminology, in this paper, so is not limited to a single word). Nouns are things and any given thing tends to have a noun describing it in each language. In contrast verbs, adverbs and adjectives tend to be tightly tied to the grammatical rules of the language and direct translations are hard. But, more than just being more tractable, nouns are sufficient for most of the motivating applications. Nouns are grouped by synonym, called a synset. We need a unique identifier for each synset.

WordNet divides nouns into 26 groups, numbered from 03 to 28, which is used as the prefix for our unique code. For instance 05 is for nouns denoting animals, and 06 is for man-made objects. So, a car, in the sense of an automobile, is in the 06 group. However a car, in the sense of a train carriage, is also in the 06 group. So a number is appended to the unique code to distinguish. Thus, 06car0 is an automobile and 06car1 is a railway car. The first noun in any given synset is used for the code. So car in the sense of cable car is 06cable_car0. WordNet drops the 0 suffix when there is only a single instance, but MLSN always has the suffix to avoid ambiguity, for instance with the word "cobalt 60" (27cobalt_600) or the movie "Die Hard 2" (10die_hard_20).

To describe relations between synsets we use 1 or 2 character symbols. So @ is hyponym, @i is instance_of, #p/#m/#s are meronyms (part_of, member_of, substance_of), ;c;/r;/u are domain terms (topic, region, usage). WordNet is similar but marks each of the parent and child slightly differently, for instance, "#p" is a part holonym, and "%p" is a part meronym. MLSN instead has a single table with parent, child and relation fields (it also a column called source, described below under legal issues). For instance, parent=06car0, child=06ambulance0, relation=@, stores a single hypernym/hyponym relation.

MLSN's goal of including culture words required adding an additional relation. P is used to describe a product-of relation. For instance: apple/i-pod/P. The alternative was to dump these relations in the general-purpose domain topic (;c) relation, but it was felt this would be losing valuable information.

The nouns DB table has the following fields: code, synonyms, gloss (dictionary definition), examples and comment. Comment is a free format field for the person viewing the data to document issues with it or to raise questions. The synonyms field has some structure. Words are comma-separated, and each word can be tagged with extra information in square brackets. This is used differently for each language, as described below. A relational database purist would have used a second table, but SQL JOINS can use a lot of CPU and memory and we are dealing with a lot of data and a lot of queries.

Further enhancements of the data structure are planned. A popularity number (measured against some large corpus) against each word would be very useful. For instance 06car0 is "car, auto, automobile, machine, motorcar". car is the most common of these synonyms for this meaning. Even more importantly the word car is found in the following synsets: 06cable_car0, 06car0, 06car1, 06car2, 06car3. When met in any random text it is overwhelmingly going to be the meaning of 06car0. This knowledge would be invaluable when

using MLSN to make search more intelligent for instance.

Another desirable enhancement is being able to flag words as, for example, formal, old-fashioned, slang, adult, British English, or even old-fashioned British English adult slang. WordNet uses the domain usage (;u) relation, but it is used inconsistently and more importantly for our multi-lingual purpose it is specific to the English language. Such knowledge belongs in the synonyms field not in the relations table.

Language-Specific Challenges

This section will describe problems inherit in the current four languages, and how the square brackets are used.

English. The square brackets are not currently used. In future the child of any ;u relation could be moved into square brackets in the synonyms field, for instance to solve the trunk/boot issue mentioned earlier.

Japanese. Furigana are the pronunciation of the word, and these are written in square brackets. They are always written in zenkaku katakana, and are used even for words that are normally written in katakana. This may seem redundant but it allows processing to be consistent and also allows us to differentiate between u and o when a long vowel follows an o sound. For instance the Japanese entry for 06cable_car0 is: "ケーブルカー [ケエブルカー], ロープウェイ [ロウブウェイ], ロープウエー [ロウブウエエ]" (The decision between ウ and オ appears to be arbitrary to a native Japanese speaker for practically all foreign borrow words; in such cases ウ has been chosen.)

In Japanese kanji can be spelled out in hiragana or katakana. Katakana can be used to emphasize a word in a similar way to the use of italics in English. Then there are noun prefixes and suffixes to change the level of formality and politeness. So we have the highly subjective issue of deciding just what is a word. Or, to put it another way, we have to know when to stop.

For instance 05wolf0 is "wolf" in English. The Japanese entry is "ウルフ [ウルフ], 狼 [オオカミ], おおかみ". Using search engine hits as a crude guide, all are in common use, with ウルフ the most popular. As another example 18father0 is: "お父さん [オトウサン], 父親 [チチオヤ], 御父さん [オトウサン], 男親 [オトコオヤ]" The inclusion of 御父さん [オトウサン] could be questioned. But what about the -sama versions? お父様 gets nearly two million search engine hits, and お父さま gets 139,000, which could be given

as evidence of them being in widespread usage. For comparison 男親 gets 138,000.

Mandarin Chinese. Conversion between simplified and traditional forms is reasonably algorithmic so simplified (as used in mainland China) is used currently, with traditional to be added at some point in the future. The square brackets store pinyin: the numeric form is used, 5 is used for no tone, and there are no spaces between the pinyin. For instance, 18mother0 is: "母亲[mu3qin1],母[mu3],妈妈[ma1ma5]".

German. Word gender is stored in square brackets. For instance, 10japanese0 is "Japanisch[n],japanische Sprache[f]". That example also shows the importance of maintaining capitalization. The controversial 1996 spelling reform gives us some challenges. The spelling of some words has been changed, for instance scharfes-S (ß) is now written as "ss". But only in some words, not all. Current MLSN policy is to include both versions on the assumption that both are likely to be in use (most schools teach the new spelling but some newspapers have bowed to public demand and reverted to traditional spelling). But in reality we are likely to only have one or the other for any given entry and not know which it is. This issue needs more attention.

Automatic Translation Of WordNet

The great hope in making a project open source is that people will flock to the project and contribute. The reality is that the project must be useful to attract users, and that only some small percentage of those users will be contributors. The catch here is obvious: without contributors it will never have enough data to be useful.

We tackle this issue using freely available dictionaries to automatically translate the English WordNet. This is very far from trivial but all is not lost as there are many words that do have simple unambiguous translations.

Our algorithm (described in MLSN 2008a in more detail) is as follows (where en refers to English, and xx refers to our target language):

1. Take all monosemous nouns from WordNet.
2. Translate each using our en-xx dictionary. Only keep those that have exactly one word in the target language.
3. Translate that one word back into English using our xx-en dictionary. Multiple words are fine. Call this set D.

4. For the original noun get its full synset from WordNet. Call this set W.

5. If D=W call it high confidence.

If D is a perfect subset of W call it medium confidence. WordNet knows more synonyms but our dictionaries did not contradict anything.

If D is not fully contained in W call it low confidence. Our dictionaries found synonyms which WordNet does not know. WordNet is reasonably comprehensive so this is suspicious.

6. Import high and medium confidence results, putting xx1h and xx1m, respectively, in the comment field. Low confidence results are thrown away.

The results are shown in table 1.

	High	Medium	Low
Japanese	5,130	7,150	3,417
German	6,795	10,085	3,540
Chinese	3,268	5,138	1,589

The decision to avoid polysemous words is a painful one, as it excludes many common words. But sticking with monosemous words gives a high level of accuracy: over 95% of high and medium choices are correct (70% of low confidence entries are correct, which was not deemed good enough for import). (See MLSN 2008a for more details.)

Another upcoming Japanese translation of WordNet (Bond, 2008) does not dodge the polysemous challenge. It uses other WordNets (French, Spanish and German), in conjunction with freely available Japanese-English/French/Spanish/German dictionaries to disambiguate. Early results show the number of high confidence automatically translated nouns (9,487) is on a par with the above algorithm, with better coverage of the most common nouns, but with lower accuracy. A detailed comparison awaits the release of the data.

About 900 words, and 1500 relations, have been manually added to the English and Japanese versions of MLSN, relative to WordNet. Another 500 entries in the Japanese version have been manually translated.

Legal Issues

Many of the data sources we use require permission to redistribute that data, or require acknowledgement. How does this affect us when we just use the data in one intermediate step of an algorithm? We bypass this question by flagging

the source of all data that is imported. Relations are simple: they are flagged with a "w" in the source field if they came from WordNet; anything else means they were added by an MLSN contributor (and the MLSN project therefore owns that data). For noun synsets the comment field is used. If the data came from WordNet (and has not been added to or changed) then it is flagged with "w". If the data came from the automatic translation process it is flagged with xx1h or xx1m where the xx is the language code. Once it has been reviewed by a human expert this flag is removed.

When the project data is released for public download only data that can be covered by the MLSN license is included. That means if the comment field contains w or xx1h or xx1m it is excluded. For relations if the source field contains w it is excluded. This is not as restrictive as it may seem - we freely make available the tools for people to reconstruct the MLSN database by downloading their own copy of the same data sources.

Harvesting Wikipedia Interwiki Links

Wikipedia has what are called interwiki links, used to link an entry in one language to the same entry in another language. For instance the English Wikipedia article for "Computer Science" has interwiki links to a Japanese article called "計算機科学", a German article called "Informatik" and a Chinese article called "计算机科学". This suggests they could be harvested to make a dictionary.

This was done (MLSN, 2008b) and found to be useful. Using just Jim Breen's JMDict (JMDict) we could automatically translate around 6,000 of WordNet's entries, with high or medium confidence. When using both JMDict and the interwiki dictionary files that figure increased to over 12,000 entries. The interwiki dictionary is also useful in that it covers lots of the culture words that are key to the MLSN project's aims.

Future Directions

The author's current effort regarding MLSN is adding more languages, the intention being to discover design problems not yet revealed by the current set of languages, so creating a strong base with which to confidently add all world languages. With this in mind the next language will be Arabic, to discover if right-to-left script, dual noun forms (in addition to singular and plural) and gender-specific nouns create problems. It is not coincidental that Arabic is currently also a language of political and economic importance. Later Korean, Russian and Spanish are planned.

References

Bond, 2008. 多言語 WordNet を利用した日本語 WordNet の作成 Francis Bond, 井佐原均, 神崎享子, 内元清貴 (2008) In 14th Annual Meeting of the Association for Natural Language Processing. Tokyo. (this volume).

Christine Fellbaum, editor, 1998. WordNet. An Electronic Lexical Database. MIT Press.

MLSN, 2006. The project home page is at <http://dcook.org/mlsn/>
Japanese home page is: <http://dcook.org/mlsn/index.ja.html>

MLSN, 2008a. Automatic Translation For MLSN, <http://dcook.org/mlsn/about/>

MLSN, 2008b. Harvesting Wikipedia Interwiki Links. <http://dcook.org/mlsn/about/>

JMDict
http://www.csse.monash.edu.au/~jwb/j_jmdict.html