LanguageWare
The Next Generation IBM Linguistic Platform

An IBM Dictionary & Linguistic Tools White Paper

This white paper introduces the reader to LanguageWare v3, the next generation IBM linguistic platform. It describes the features delivered in this new product and is written for anyone who has an interest in the exploitation of linguistic processing.

Marie Wallace
Product Manager
Friday, October 11, 2002

Introduction

To understand the competitive advantage that LanguageWare delivers it is important to understand the new paradigm that is being proposed. The LanguageWare paradigm was conceived through questioning the basis under which linguistic functions had traditionally been deployed, and in parallel performing a detailed analysis of the usage scenarios behind the creation of natural language processing (NLP) applications. The result of which was a radical shift in the way in which we considered the infrastructure that we would deliver to customers.

We observed that most linguistic solutions tend to provide functions via a “bag of tools” which can be arbitrarily called to deliver a series of linguistic results. The functions correspond to different levels of linguistic processing, for example, preprocessing, tokenization, lemmatization or POS tagging. Each layer of function calls adds application overhead.

In LanguageWare we have tried to take a more holistic view of linguistic processing. We have architected our solution to seamlessly integrate our linguistic processing capabilities with our finite-state dictionaries. We have used our dictionaries to efficiently store information relating to several levels of linguistic processing through the development of multiple gloss types in the one dictionary. Within our dictionaries we can handle various data-types; morphological, morphosyntactic, semantic, pragmatic, and user-defined. There is no need to have separate formats and dictionaries for different types of data, such as word-frequencies for Chinese, word-formation elements in German, or stopword lists used in information retrieval.

This is why we can, through just one dictionary lookup, provide you all possible information associated with the lexical units.

Therefore, in summary the LanguageWare paradigm is based on the premise that, with the efficient consolidation of various types of dictionary data and using smart dictionary construction, only one dictionary lookup is ever required as part of lexical analysis. In addition, this lookup allows customers to access, without any cost in performance, all other data (glosses) associated with the resulting lexical units. This data might be the lemma, a stopword flag, morphosyntactic data, perhaps some statistical probabilities, or any type of user-defined data. Whatever information the application could possibly associate with lexical units can be exploited through our LanguageWare solution.

LanguageWare Dictionary

Structure

In order to successfully implement the LanguageWare paradigm we needed to consolidate various types of lexical data within our dictionaries. We did this through constructing each entry as consisting of an index, a string of characters representing a word or a phrase, and a piece of associated information. This associated information is represented as a gloss collection, which consists of a series of glosses of various types. There are a number of default gloss types that are defined within LanguageWare, such as lemma, part-of-speech (POS), morphology, synonym, or stopword, however more importantly is the ability to register user-defined gloss types which can be used to store any information. This user-defined gloss type can have any internal structure, binary or otherwise, the assumption being that the user who created this data will process it within their own application. LanguageWare simply provides the interface to allow access to this data, extremely quickly and using the generic LanguageWare dictionary lookup interface.

For example, a summarization algorithm might find it useful to tag certain words or expressions as being “change of discourse”. Another application might find statistical information important during its processing. Therefore, with LanguageWare instead of having to create multiple dictionaries and perform many dictionary lookups you can access all this information transparently during lexical analysis.

UniMorph

Access to morphosyntactic information within the dictionaries is now facilitated through a new interface called UniMorph. UniMorph is based on a study of morphosyntactic phenomena. The study identified common and unique features across the main Western European languages and described these in a language neutral framework. We have taken this basis and extended it to more accurately represent our language/data set and our customer requirements.

Essentially, what UniMorph provides is an interface that is identical across all languages and which allows you to easily identify morphosyntactic features associated with the lexical units. Due to the language neutral nature of the interface it makes it relatively transparent to extend functionality across many languages.

Content

LanguageWare ships with a number of dictionaries for each language. There is the primary language dictionary which contains the main word list for the respective language. It contains information such as lemma, part of speech (POS), morphology and stopword flags, and is used during lexical analysis. In addition, for most languages there is a synonym dictionary, abbreviation dictionary and in some cases a multi-word expression dictionary. Finally, there are a number of supplementary

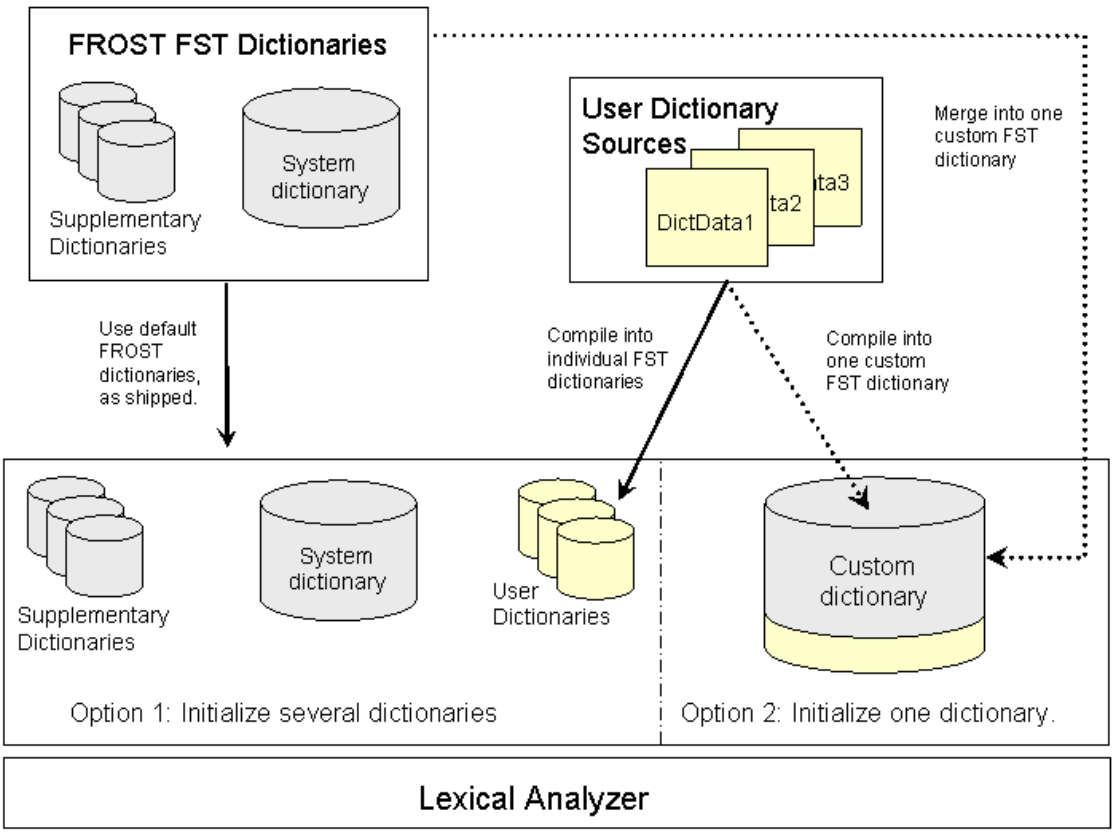
dictionaries which contain such information as IBM terminology, and medical, computer, or legal terms.

Customization

While the LanguageWare dictionaries may be adequate for many customers, there will be those that want the ability to customize the dictionaries to suit their own specific requirements. LanguageWare dictionary customization allows customers to develop their own dictionary data and then either use the dictionary independently within the LanguageWare system or merge it with any number of other dictionaries to create their own specialized custom dictionary. You have complete control over your own dictionary data and simply re-merge whenever you want your changes to get re-integrated into the core LanguageWare data. Likewise when any updates to LanguageWare data are available all you have to do is re-merge with your dictionaries in order to get the updates.

The benefit of this merge facility at runtime is that one integrated dictionary allows you to take advantage of all dictionary data seamlessly and without any performance overhead.

The dictionary customization functionality is more clearly demonstrated in the diagram below.



Lexical Analysis

Lexical Analysis is the framework on which the LanguageWare solution is built and uses a combination of dictionary lookup and algorithmic processing to segment input text into distinct lexical units. Automatically, without any additional function overhead, it provides access to the gloss information associated with these lexical units.

The current version of the Lexical Analyzer supports the accurate segmentation for the following linguistic phenomena.

1. Word segmentation for Japanese and Chinese.
2. Contractions – slit split into their component parts. For example, wouldn't -> would + not, Horse's -> Horse + is/'s (ambiguity).
3. Clitics – split into their component parts. For example, reparti-lo-emos -> repartir + lo + emos, l'avenue -> le + avenue, dell'arte -> dello + arte.
4. Compounds – split into their component parts. For example, Oberschulrat -> Ober + Schule/schulen (ambiguity) + Rat/raten (ambiguity)...
5. Multi-word expressions – if the expression is lexicalized (in the dictionary) it will be recognized as one lexical unit. For example, International Business Machines, tip of the iceberg, George W. Bush, etc.
6. Abbreviations – if an abbreviation is lexicalized it will be recognized as one lexical unit. If it is not lexicalized it will still be recognized as a lexical item, however will not have any associated gloss information.
7. End of sentence (EOS) markers – a basic level of EOS detection is performed against punctuation.
8. Non-alphabetic characters – due to the internal use of ICU character Iterator, all non-alphabetic characters are returned as separate lexical units. The exception to this is cases where LanguageWare has additional logic to deal with them differently. For example, apostrophes or hyphens in the case of clitics, full-stops in the case of unknown abbreviations, or any non-alphabetic characters where they are explicitly lexicalized.

It is important to note that LanguageWare does not currently provide part of speech (POS) disambiguation and therefore we pass all ambiguities back up to the calling application.

Synonym Lookup

Synonym lookup is facilitated through the generic LanguageWare dictionary lookup function and the LanguageWare synonym dictionaries that ship with the product. A synonym lookup will return you the synonym gloss which contains a list of synonyms. Like with the other dictionaries you can customize the synonym dictionary to modify synonym entries.