Semantic Web Ontology for reasoning over the phonological concepts related to letters in alphabetic languages

Gina (Virginia) Cook
5918626 ginacook@alumni.concordia.ca

Abstract

Normally, languages (such as most Latinate and Germanic Languages) do not include predictable information in the orthography. However, some writing systems (Turkish, Finnish, Inuktitut) include predictable and systematic information which causes an exponential increase of vocabulary size. This introduces problems of computation time and complexity when doing any Natural Language Processing task such as Information Retrieval. Fortunately some of this predictable information is well understood by phonologists who work on these languages. This paper discusses an ontology which encodes the phonological, phonetic and orthographic concepts and relations which a Phonologist reasons over to discover the predictable systematicities in a languages spoken or written stream. More importantly this ontology can operate on any UTF-8 formatted text and use case based resoning to return possible systematicities which can be investigated and removed from the vocabulary set to reduce the vocabulary size.

1 Introduction

This paper serves to document the methods, results, and lessons learned in attempting to build a phonological ontology (Phonont.v1) which can be populated with any UTF-8 text data and provide some meaningful query services. The ontology's Abox was built and tested using UTF-8 data from a variety of languages of a variety of corpus size and type. The ontology's Tbox is composed of two primary knowledge sources, one prepared mostly by hand from literature on Phonological Theory, the other automatically extracted from the corpora. Wherever possible information was automatically generated so as to provide cyclic data processing to discover and capitalize on systematicities in human reasoning when accumulating and integrating knowledge.

2 The Problem - Why model the ontology of phonological concepts

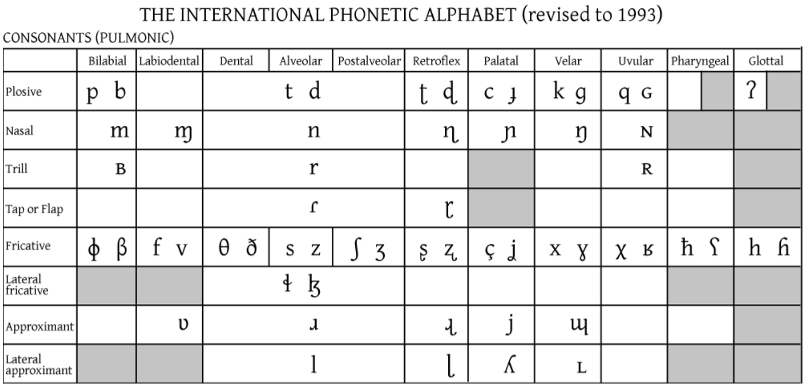
The overarching goal behind Phonont.v1 is to explore both the possibilities and potential roadblocks in Ontological modeling (§ 2.1), its secondary goal is to understand ways of automating a phonological reasoner for tasks in Natural Language Processing which would benefit from removing phonological systematicities which cannot be detected based on letter counting alone (§ 4.1).

2.1 Lessons learned about the Semantic Web

The Semantic Web and ontologies or graph representations in general are very natural ways of encoding Taxonomic relations between concepts (such as *(/p/ isA Bilabial*), as well as more "meaningful" relations between concepts (such as *(/p/ isSimilarTo /b/*).

2.1.1 Roles vs. Concepts

Phonont.v1 uses a great deal of Taxonomic relations, as this is traditionally how phonological features are conceptualized as shown in the IPA table (fig. 1) where the "concepts" are found along the row and column labels and "individuals" are found in the intersection of these concepts. As can immediately be seen from this table, in fact the cells of the table may be represented as concepts, and the actual words of a language could be used as individuals. This can be yet further zoomed to consider the words of the language to be concepts, and the individual utterances as individuals. The level of zoom largely depends on the task. For speech recognition words should be concepts and acuoustic recordings of the words can be individual instantiations of the word. For Phonological reasoning it might be prefered for letters to be concepts and words to represent the individual uses of the letter in context.



THE INTERNATIONAL PHONETIC ALPHABET (revised to 1993)

CONSONANTS (PULMONIC)

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap				ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Figure 1: International Phonetic Alphabet is organized using a Table which hides a more Ontologically complex network of concepts

In the course of modeleing phonological concepts it became clear that the table in fig. 1 is an oversimplification of the concepts involved in human speech sounds. Phonont.v1 aims to unify concepts from various Phonological frameworks (SPE, Halle and Chomsky 1968, Feature Geometry Vaux 2000, ArticulatoryPhonetics, Keating and Lahiri 1989, Goldsmith 1990). Most of the discoveries and lightbulb moments involved in trying to define and unify concepts would only be interesting to Phonologists, so they will not be discussed here. A section of the inferred model of the hand built Phonology Tbox which is part of Phonont.v1 is shown in (fig. 2). This process of attempting to unify concepts across frameworks serves to highlight research questions which are being asked, and also provide the conceptual structure to make queries of documents to better understand these questions.

2.1.2 Nominals

How to model nominals (components which can be modeled as either individuals or concepts) a topic which can greatly increase the time complexity of a Semantic Web Reasoner. The RacerPro reasoner used in Phonont.v1 does not support nominals (or rather, approximates nominals). For both the reasons of complexity and support, nominals were not used in Phonont.v1. Where to draw the line in the zoom is a tricky question. After some experimentation with different levels of zoom it was decided to use words in a corpus as individuals and letters as concepts. After running some queries with Racer this choice became less clear. It would have been perhaps easier to quiery if letters were made to be individuals.

The choice of encoding an idea as a concept, *LettersSimilarToP:* $\{b,m,t,k\}$ or as a role *p isSimilarTo b, p isSimilarTo* $\{m\ldots\}$ is directly related to the types of individuals available in the system. I struggled to not use letters as individuals so that I could use roles to connect them. As shown in fig. 3 Individuals of a letter class were represented by a sample word from the corpus, in this case The Inuk Magazine corpus of Romanized Inuktitut (Inuktitut in Roman letters). The Semantic Web does not allow roles to connect concepts. The only roles that can concept concepts are 'taxonomic' roles, refered to as subsumption, but is similar to ideas in other domains such as *isA, isPartOf, isATypeOf, isAKindOf, impiles, isSubsumedBy*.

2.1.3 Ontologies and Lattices

In their discussion on using Ontologies to encode publications and research Tho et a. 2006 merge the ideas of Clustering and Fuzzy Logic in Text Classification with the TBox and Abox ideas of Semantic Web. Their formalisms for the Fuzzy Ontology Generation frAmework (FOGA) are very reminicent of joined semi-latices used in Set Theory Semantics (Link 1983), an approach which uses sets rather than truth values to calculate the meaning of sentences. In fact, Ontologies are much like latices, where sets are connected via the $\sqsupseteq$ operator, however not all sets are connected to other sets. When the individuals of a Ontology are viewed this way it can become clear why additional individuals cause exponential growth in classification time. The lattices in fig. 4 show the transitivity of the implication/entailment/subset relationship. The lattices also show why it is important to have an individual asserted in a concept if that concept must be used in reasoning. Furthermore, the figure shows the difference between 4 individuals

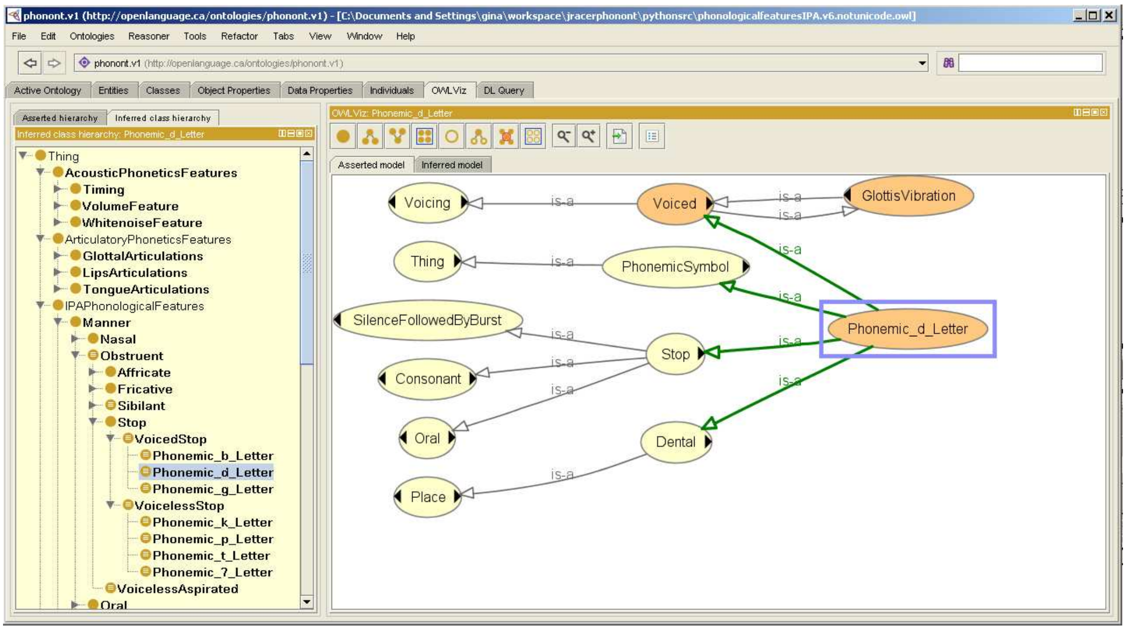

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1) - [C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\phonologicalfeaturesIPA.v6.notunicode.owl]
File Edit Ontologies Reasoner Tools Refactor Tabs View Window Help
phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1)
Active Ontology Entities Classes Object Properties Data Properties Individuals OWLViz DL Query
Asserted hierarchy Inferred class hierarchy
Inferred class hierarchy: Phonemic_d_Letter
Thing
AcousticPhoneticsFeatures
Timing
VolumeFeature
WhitenoiseFeature
ArticulatoryPhoneticsFeatures
GlottalArticulations
LipsArticulations
TongueArticulations
IPAPhonologicalFeatures
Manner
Nasal
Obstruent
Affricate
Fricative
Sibilant
Stop
VoicedStop
Phonemic_b_Letter
Phonemic_d_Letter
Phonemic_g_Letter
VoicelessStop
Phonemic_k_Letter
Phonemic_p_Letter
Phonemic_t_Letter
Phonemic_?_Letter
VoicelessAspirated
Oral
OWLViz: Phonemic_d_Letter
Asserted model Inferred model
Voicing
Voiced
GlottisVibration
Thing
PhonemicSymbol
SilenceFollowedByBurst
Phonemic_d_Letter
Stop
Consonant
Oral
Dental
Place
is-a

Figure 2: Phonological Ontology provides opportunities to unify and ask questions about the distinction between concepts across phonological frameworks

and 7, wher more individuals cause exponentially more potential connections in the graph and cause longer compute times.

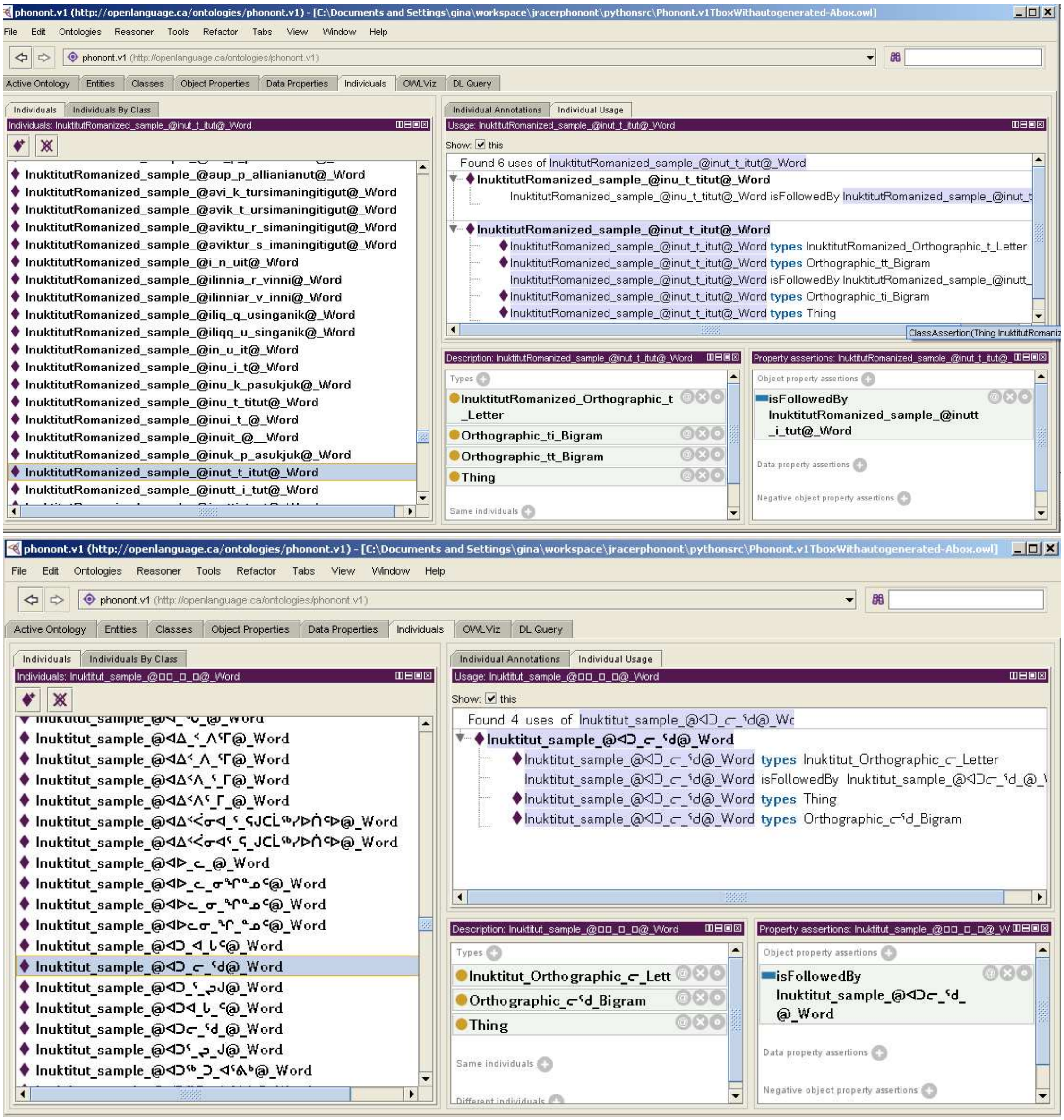
2.1.4 Ontologies and Uncertainty

One of the most important problems in building a Phonological reasoner is reasoning with uncertain edges of concepts. This has been identified as a research area in Semantic Web. Much of Human reasoning includes likleyhoods of truth, or degrees of membership such as very expensive, hot beach (what does hot mean?) (Stoilos et al. 2005). There are a few approaches to uncertainty, including Baisian Nets and Fuzzy Logic.

One of the tasks in building Phonont.v1 was to populate the ontology with letters of the corpus. This feature of Phonont.v1 can later be used in Language Recognition to determine the Language source of words in a document which is multi-lingual (a rising problem in most languages where English words are sprinkled among native words). A fuzzy logic implementation of “coreness” was used to encode this valuable information. After running the Python script which extracts letter counts from texts, letter frequencies were graphed and visualized using GnuPlot (fig. 5). Letters from other languages (such as French letters appearing in English or Inuktitut texts) appeared in the bottom tail of the letter frequencies. Letters which appeared less than 1% of the most frequent letter were labled as NonCore letters for the language of that Corpus. This formalism worked well even for corpora that were rather ‘clean’ of foreing letters such as the phonologically transcribed Brent English corpus (discussed in § 3.1). Letters which appeared in the top 30% were labeled as VeryCore (shown in fig. 6 VeryCore letters could serve in langauge recognition but also later in in determining which phonotactic clusters should be present but are missing from the data.

2.1.5 The Semantic Web and Internationalization

The vast majority of man hours behind the Phonont.v1 development went into dealing with quirks of text encoding. Both in software applications, and XML and the corpora sources themselves (the Inuktitut Corpus was generated by spidering two different websites then transcoding the information). RacerPro 2.0 now supports UTF-8 unicode encoding. Jracer, implemented in Java also supports UTF-8. Protege 4 also supports UTF-8 encoding. Python also supports UTF-8 but will complain profusely when treating ASCII as Unicode. Yet, mysteriously off and on data would become unreadable, possibly due to developing between multiple machines, operating systems, text editors



phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1) - [C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\Phonont.v1TboxWithautogenerated-Abox.owl]

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1)

Individuals: InuktitutRomanized_sample_@inut_t_itut@_Word

InuktitutRomanized_sample_@aup_p_allianianut@_Word
InuktitutRomanized_sample_@avi_k_tursimaningitigut@_Word
InuktitutRomanized_sample_@avik_t_ursimaningitigut@_Word
InuktitutRomanized_sample_@aviktu_r_simaningitigut@_Word
InuktitutRomanized_sample_@aviktur_s_imaningitigut@_Word
InuktitutRomanized_sample_@i_n_uit@_Word
InuktitutRomanized_sample_@ilinnia_r_vinni@_Word
InuktitutRomanized_sample_@ilinniar_v_inni@_Word
InuktitutRomanized_sample_@iliq_q_usinganik@_Word
InuktitutRomanized_sample_@iliqq_u_singanik@_Word
InuktitutRomanized_sample_@in_u_it@_Word
InuktitutRomanized_sample_@inu_i_t@_Word
InuktitutRomanized_sample_@inu_k_pasukjuk@_Word
InuktitutRomanized_sample_@inu_t_titut@_Word
InuktitutRomanized_sample_@inui_t_@_Word
InuktitutRomanized_sample_@inuit_@__Word
InuktitutRomanized_sample_@inuk_p_asukjuk@_Word
InuktitutRomanized_sample_@inut_t_itut@_Word
InuktitutRomanized_sample_@inutt_i_tut@_Word

Usage: InuktitutRomanized_sample_@inut_t_itut@_Word

Show: ☑ this

Found 6 uses of InuktitutRomanized_sample_@inut_t_itut@_Word
InuktitutRomanized_sample_@inu_t_titut@_Word
InuktitutRomanized_sample_@inu_t_titut@_Word isFollowedBy InuktitutRomanized_sample_@inut_t
InuktitutRomanized_sample_@inut_t_itut@_Word
InuktitutRomanized_sample_@inut_t_itut@_Word types InuktitutRomanized_Orthographic_t_Letter
InuktitutRomanized_sample_@inut_t_itut@_Word types Orthographic_tt_Bigram
InuktitutRomanized_sample_@inut_t_itut@_Word isFollowedBy InuktitutRomanized_sample_@inutt_
InuktitutRomanized_sample_@inut_t_itut@_Word types Orthographic_ti_Bigram
InuktitutRomanized_sample_@inut_t_itut@_Word types Thing

ClassAssertion(Thing InuktitutRomaniz

Description: InuktitutRomanized_sample_@inut_t_itut@_Word

Types
InuktitutRomanized_Orthographic_t_Letter
Orthographic_ti_Bigram
Orthographic_tt_Bigram
Thing
Same individuals

Property assertions: InuktitutRomanized_sample_@inut_t_itut@_

Object property assertions
isFollowedBy InuktitutRomanized_sample_@inutt_i_tut@_Word
Data property assertions
Negative object property assertions

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1) - [C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\Phonont.v1TboxWithautogenerated-Abox.owl]

phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1)

Individuals: Inuktitut_sample_@□□_□_□@_Word

Inuktitut_sample_@ᐊᐃ_ᑉ_ᐱᕐᒥ@_Word
Inuktitut_sample_@ᐊᐃᑉ_ᐱ_ᕐᒥ@_Word
Inuktitut_sample_@ᐊᐃᑉᐱ_ᕐ_ᒥ@_Word
Inuktitut_sample_@ᐊᐃᑉᐱᕐ_ᒥ_@_Word
Inuktitut_sample_@ᐊᐃᑉᐹᓂᐊ_ᕐ_ᕆᒍᑕᒫᖅᓯᐅᑏᕐᐅ@_Word
Inuktitut_sample_@ᐊᐃᑉᐹᓂᐊᕐ_ᕆ_ᒍᑕᒫᖅᓯᐅᑏᕐᐅ@_Word
Inuktitut_sample_@ᐊᐅ_ᓚ_@_Word
Inuktitut_sample_@ᐊᐅ_ᓚ_ᓂᖕᒋᓐᓄᑦ@_Word
Inuktitut_sample_@ᐊᐅᓚ_ᓂ_ᖕᒋᓐᓄᑦ@_Word
Inuktitut_sample_@ᐊᐅᓚᓂ_ᖕᒋ_ᓐᓄᑦ@_Word
Inuktitut_sample_@ᐊᑐ_ᐊ_ᒐᑦ@_Word
Inuktitut_sample_@ᐊᑐ_ᓕ_ᖁ@_Word
Inuktitut_sample_@ᐊᑐ_ᕐ_ᓱᒍ@_Word
Inuktitut_sample_@ᐊᑐᐊ_ᒐ_ᑦ@_Word
Inuktitut_sample_@ᐊᑐᓕ_ᖁ_@_Word
Inuktitut_sample_@ᐊᑐᕐ_ᓱ_ᒍ@_Word
Inuktitut_sample_@ᐊᑐᖅ_ᑐ_ᐊᕐᕕᒃ@_Word

Usage: Inuktitut_sample_@□□_□_□@_Word

Show: ☑ this

Found 4 uses of Inuktitut_sample_@ᐊᑐ_ᓕ_ᖁ@_Wc
Inuktitut_sample_@ᐊᑐ_ᓕ_ᖁ@_Word
Inuktitut_sample_@ᐊᑐ_ᓕ_ᖁ@_Word types Inuktitut_Orthographic_ᓕ_Letter
Inuktitut_sample_@ᐊᑐ_ᓕ_ᖁ@_Word isFollowedBy Inuktitut_sample_@ᐊᑐᓕ_ᖁ_@_\
Inuktitut_sample_@ᐊᑐ_ᓕ_ᖁ@_Word types Thing
Inuktitut_sample_@ᐊᑐ_ᓕ_ᖁ@_Word types Orthographic_ᓕᖁ_Bigram

Description: Inuktitut_sample_@□□_□_□@_Word

Types
Inuktitut_Orthographic_ᓕ_Lett
Orthographic_ᓕᖁ_Bigram
Thing
Same individuals
Different individuals

Property assertions: Inuktitut_sample_@□□_□_□@_W

Object property assertions
isFollowedBy Inuktitut_sample_@ᐊᑐᓕ_ᖁ_@_Word
Data property assertions
Negative object property assertions

Figure 3: Example of individuals and Relations in Phonont.v1

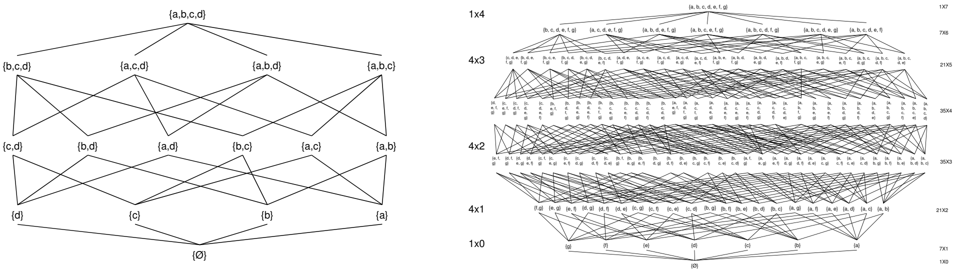


Figure 4: Truth can be viewed as the presence or absence of members in the set. Compare the complexity of fully connected lattices with 4 individuals and a lattice with 7.

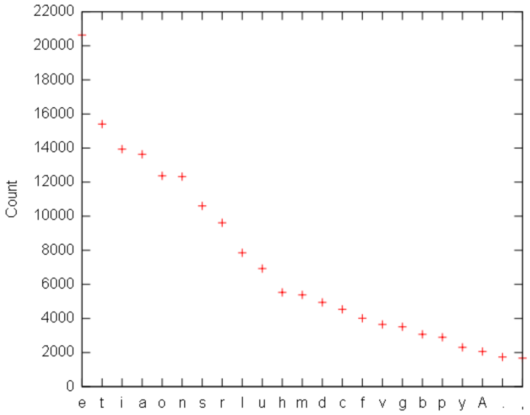
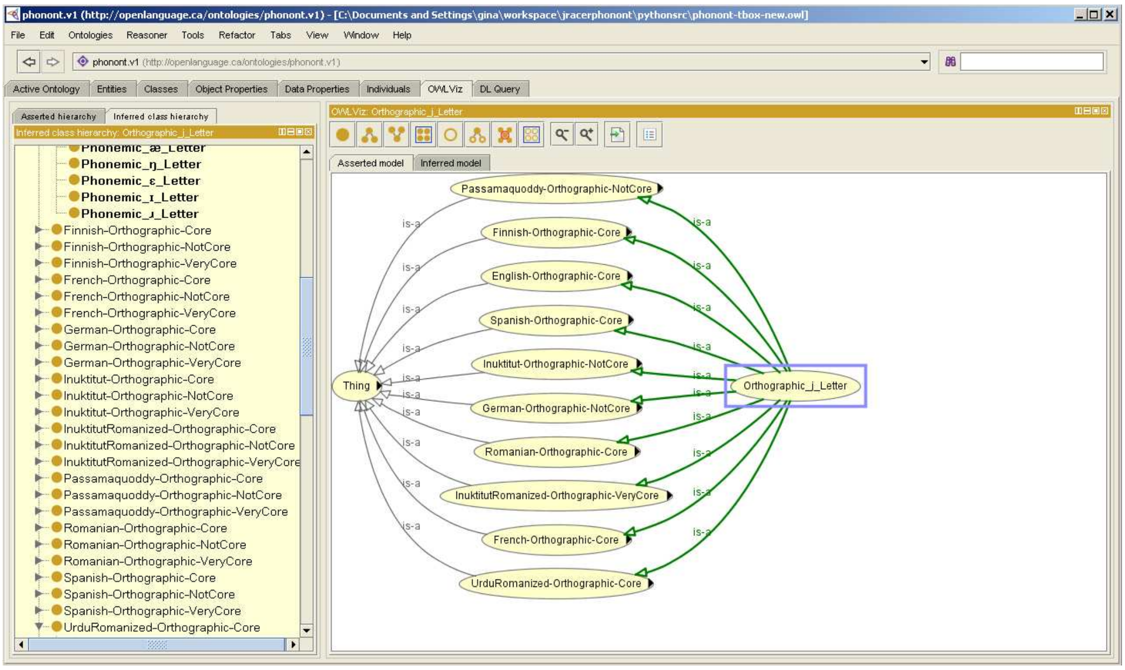


Figure 5: Sample graph of Zipf's law for letter frequencies, used to create Fuzzy sets of VeryCore, Core and NotCore letters per language.


phonont.v1 (http://openlanguage.ca/ontologies/phonont.v1) - [C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\phonont-tbox-new.owl]

Phonemic_æ_Letter
Phonemic_ŋ_Letter
Phonemic_ɛ_Letter
Phonemic_ɪ_Letter
Phonemic_ɹ_Letter
Finnish-Orthographic-Core
Finnish-Orthographic-NotCore
Finnish-Orthographic-VeryCore
French-Orthographic-Core
French-Orthographic-NotCore
French-Orthographic-VeryCore
German-Orthographic-Core
German-Orthographic-NotCore
German-Orthographic-VeryCore
Inuktitut-Orthographic-Core
Inuktitut-Orthographic-NotCore
Inuktitut-Orthographic-VeryCore
InuktitutRomanized-Orthographic-Core
InuktitutRomanized-Orthographic-NotCore
InuktitutRomanized-Orthographic-VeryCore
Passamaquoddy-Orthographic-Core
Passamaquoddy-Orthographic-NotCore
Passamaquoddy-Orthographic-VeryCore
Romanian-Orthographic-Core
Romanian-Orthographic-NotCore
Romanian-Orthographic-VeryCore
Spanish-Orthographic-Core
Spanish-Orthographic-NotCore
Spanish-Orthographic-VeryCore
UrduRomanized-Orthographic-Core

Passamaquoddy-Orthographic-NotCore
Finnish-Orthographic-Core
English-Orthographic-Core
Spanish-Orthographic-Core
Inuktitut-Orthographic-NotCore
Thing
Orthographic_j_Letter
German-Orthographic-NotCore
Romanian-Orthographic-Core
InuktitutRomanized-Orthographic-VeryCore
French-Orthographic-Core
UrduRomanized-Orthographic-Core
is-a

Figure 6: Fuzzy Set Membership can serve to encode relative "importance" of automatically generated data.

and command prompt/IDE environments. Ultimately most of the development was done in Eclipse on Windows as getting RacerPorter to run, and JRacer to run was rather straightforward. After prolific testing of JRacer with various UTF-8 texts to successful results, development went ahead using Eclipse, Java and Protege to build the Phonont Tboxes and Aboxes.

After much development it was discovered that source of continuous errors of perfectly valid UTF-8 and OWL, were due to XML. According to errors in Protege, XML does not allow (punctuation specific?) UTF-8 in comments, more specifically in the URI which expresses the location of resources in OWL. Unfortunatly puctuation and digits are used in some texts to encode a phonological sound. As a case in point, the Passamaquoddy corpus (§ **??**) uses the letter 4 to indicate a glottal stop, and much chat language also uses punctuation or digits for encoding speech so this information should not be thrown away apriori from the Phonont as it is supposed to deal with any language, as long as it is in UTF-8 encoding. Keeping as much punctuation and digits as possible in order to discover these facts caused cycles of Protege errors, editing the Python regular expressions, checking the original corpus text, running the model again. By the end of development the pruning of punctuation and UTF-8 characters which the XML objected to, would take over 20 minutes to test while Phonont was being generated. In a future implementation I will explore writing the Ontology in something other than XML. I have considered using Racer's language, and creating a separate OWL XML generator which operates on this. As the code behind Phonont is modular this requires only changing one line in the population script, and perhaps adding a housekeeping array to declare all entitites in the ontology.

3 Generating the Ontology

The ontology can be generating using the Java Class Text2Ontology.java included in the source of the ontology at http://openlanguage.ca/ontologies/phonont.v1/ The Java class begins by calling several Python scripts which extract and count words in the corpora, extract and count letters, and bigrams, and finally populate mini-Tbox/aboxes using the letters, bigrams and words for each corpus. These Tboxes can then be later selected for addtion into the base TBox, along with the appropriate phoneme mapping for that language (also automatically generated by TBoxConstruction.java).

3.1 Knowledge Extraction

When the entire corpus is run and populated there are over 15000 assertions and it takes over 20 minutes to generate. Of course, normally only the Tboxes are needed in the final Phonont, the Aboxes can be added when the Ontology is being queried about a particular set of languages. Below is the output of Text2Ontology.java to demonstrate the control flow of the Ontology generation process. In addition to the corpora included in the base Tbox below there is also an Abox and Tbox for Korean, however given its syllabic orthography generates over 1000 classes in just the Tbox alone, compared to most other languages which range between 20-40 concepts for their Tbox (most of which overlap with the other European languages). Below is a discussion the various languages and corpora tested with Phonont, as well as brief explanations of why testing various languages and corpora is important for unsupervised Natural Language Processing. It is important to note that much of what needs to be discovered is either Statistical or Ontological, for this reason an Ontology like Phonont stands to make a contribution to accuracy benchmarks.

Language is English

File is br-phono.txt-unicode.txt

Remarks: This is a corpus of phonemically transcribed English Speech. The sentences are very short, as it is from the adult speech directed towards children. This corpus is used by Natural Language Processing Researchers to "learn" to segment English based on phonotactics. It is considered the closest possible representation to what is in English speakers heads rather than the product of a writing system. This corpus is very 'clean' there are no extra symbols. Interestingly, the Fuzzy sets used in Phonont is able to detect this fact, in that no semgents are included as NotCore. Other corpora all have NotCore letters, usually punctuation, foreign letters and some rare letters which often are borrowed (ie. the letter k in french is rare, and found mostly in borrowed words from Germanic langauges.)

(' Total words in file: ', 33399)

(' Most Frequent letter ,', 353)

(' Most Frequent bigram ,', 904)

Language is German

File is german-mommsen-Roemische_Geschichte-Book01.utf8.txt

(’ Total words in file: ’, 86664)

(’ Most Frequent letter ,’, 12995)

(’ Most Frequent bigram ,’, 52309)

Language is English

File is english-1_Harry_Potter_and_the_Sorcerers_Stone.txt

(’ Total words in file: ’, 77601)

(’ Most Frequent letter ,’, 4048)

(’ Most Frequent bigram ,’, 8466)

Language is French

File is french-1_Harry_Potter_et_la_Pierre_Philosophale.utf8.v2.txt

(’ Total words in file: ’, 85889)

(’ Most Frequent letter ,’, 6290)

(’ Most Frequent bigram ,’, 13900)

Language is Spanish

File is spanish-1_Harry_Potter_y_la_Piedra_Filosofal-cleaned.utf8.txt

(’ Total words in file: ’, 78177)

(’ Most Frequent letter ,’, 5915)

(’ Most Frequent bigram ,’, 13231)

Language is Passamaquoddy

Remarks: This is a corpus of stories written in a strange transcription code which makes use of ‘ to symbolize palatalization and 4 to sybolize a glottal stop. An interesting example of othographies which must be considered when doing Natural Language Processing. Similar othographies include Romanized Arabic and SMS/Chat language corpora in most languages. This is important to consider when building autocomplete features on cellphones or on websites.

File is passamaquoddy-TalesFromMaliseet.utf8.txt

(’ Total words in file: ’, 17393)

(’ Most Frequent letter ,’, 2787)

(’ Most Frequent bigram ,’, 12060)

Language is UrduRomanized

Remarks: Like many romanized scripts Romanized urdu is non-standard, which means users in different areas, or from differnet internet communities will develop their own orthographies. Phonont can help determine “similarity” of spellings to discover that /hai/ and /he/ might be alternative spellings of the same word as it is sound-similarity which drives spellers choices.

File is urdu-forum-060427aagar.aap.ke.pechey.kotta.lag.jaey.v4-notimes.txt

(’ Total words in file: ’, 5987)

(’ Most Frequent letter ,’, 1155)

(’ Most Frequent bigram ,’, 1493)

Language is Romanian

File is romanian-ziare.0.v5sentences-cleaned.utf8.txt

(’ Total words in file: ’, 339506)

(’ Most Frequent letter ,’, 23506)

(’ Most Frequent bigram ,’, 51121)

Language is InuktitutRomanized

File is InukMagazine102-104rough-inuktitut.utf8.txt

(’ Total words in file: ’, 17874)

```
(’ Most Frequent letter ,’, 9293)
(’ Most Frequent bigram ,’, 42402)
```

Language is Inuktitut

Remarks: This is a corpus of syllabic Inuktitut taken from the Nunavut Assembly news website. The data is suspicious in that there are some sentences which contain no spaces, resulting in bizare word choices and unrepresentatitive bigrams which span arcorss word boundaries in ways that are not really in the language. For this reason on exciting results can be gleaned from this corpus, and the Romanized Inuktitut must be used until I make a suitable Inuktitut Corpus.

```
File is assembly.nu.news-innuktitut.txt
(’ Total words in file: ’, 9764)
(’ Most Frequent letter ,’, 2890)
(’ Most Frequent bigram ,’, 40762)
```

Language is Finnish

Remarks: Finish and Turkish were tested to develop queries for Vowel harmony. Vowel harmony means that vowels within a word should ahve the same feature for “Round.” This is different from German and French where each individual vowel is either Round or Unround. This is the sort of discover which is best done with an ontology in that a query can be made to retreive the features of the vowels with in a word, if the vowel features often match, then the Phonont can ‘discover’ vowel harmony. There are other types of harmony (palatization, aspiration, tones) but vowel harmony is rather prevalent in the alphatbetic orthographies of the world and is thus an important thing to ‘discover’ automatically.

```
File is leipzig-fi.small.txt
(’ Total words in file: ’, 406791)
(’ Most Frequent letter ,’, 82688)
(’ Most Frequent bigram ,’, 307525)
(’ The number of assertions written to file: ’, 15512)
```

4 Competency Questions

4.1 What can be extracted from Aboxes?

Phonont is populated with sample individuals (words) from texts. It can do case based reasoning to provide potential answers to questions in text processing like which words have a certain sound (fig. 7A), which language is this word (fig. 7B), and what sounds does the Inuktitut letter /p/ appear next to, as shown in (fig. 7C).

4.2 What can be extracted from Tboxes?

4.2.1 Allophonic Rules

English speakers usually think they are saying a ’s’ (a *phoneme* is defined as a mental representation of a speech sound in one’s language) in the words /statement**s**/ and /algorithm**s**/, when they might be pronouncing a ’s’ [statement-s] or ’z’[algorithm-z] (*allophones* are defined as the variant realizations of a phoneme when produced) depending on the sounds around it. Given that speech segments are created in physical space the muscles involved are not independent and as a result a speech sound will be produced differently depending on the sounds around it.

Allophones are always “similar” to their underlying mental representation, usually differing in one phonological or phonetic feature. For example, [s] voiceless dental fricative, and [z] voiced dental fricative, differ only in voicing. As another example, the most significant contribution to a Spanish speaker’s accent in English (and what makes it difficult to detect which consonant they are producing) is the phonological rule that all voiced stops (/d/,/b/,/g/) are produced as voiced fricatives, which have a much shorter duration and are much less distinct from sourronding sounds. This is modeled in Phonont using the Phonetic features. If Phonont is populated with a Spanish orthography-phoneme mapping and Spanish data this query can be answered. The data for this paper focuses on English and Inuktitut as representatives of vastly different systems. English orthography reflects almost nothing of the pronunciaiton, while Romanized Inuktitut reflects almost perfectly (being a new orthographical system) the pronuncation of words. This can be quieried by using links between concepts. In the Semantic Web the only role between concepts is that of subsumption. This can be easily extracted from the OWL source. While Racer focuses on quering individuals, most phonological inference is between concepts.

```
RACER OUTPUT:
Racer Message (STDOUT):
Reading ontology C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\Phonont.v1TboxWithautogenerated-Abox-SampleInuktitutEnglish.owl...
Reading ontology C:\Documents and Settings\gina\workspace\jracerphonont\pythonsrc\Phonont.v1TboxWithautogenerated-Abox-SampleInuktitutEnglish.owl done.


Racer Message (STDOUT):
Classifying TBox......................................................
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
...

Let’s use Phonont to do some Case Based reasoning, ie, query individuals in the Abox to find potential answers,
if the individual isnt there, that doesn’t the answer is extensive, there might be individuals that are not in the
 abox that could serve as important information...
A.) Which words in which abox languages have the phoneme j, palatal approximate?
ple Case:     x
Example Case:  InuktitutRomanized\_sample\_@nunaqaqqaaqsima\_j\_ut@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@qaritau\_j\_akkut@\_Word       x
Example Case:  English\_sample\_@the\_y\_@\_Word       x
Example Case:  English\_sample\_@empt\_y\_@\_Word       x
...
B.) There is a word in a text with th, which language might it be?
ple Case:     x
Example Case:  English\_sample\_@\_t\_he@\_Word       x
Example Case:  English\_sample\_@t\_h\_e@\_Word
C.) Where can the letter p occur in Inuktitut?
ple Case:     x
Example Case:  InuktitutRomanized\_sample\_@Ta\_p\_iriit@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@\_p\_ijjutigillugu@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@sivullir\_p\_aami@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@kanatau\_p\_@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@pijunnautiqaq\_p\_ut@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@inuk\_p\_asukjuk@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@tisi\_p\_iri@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@au\_p\_pallianianut@\_Word       x
Example Case:  InuktitutRomanized\_sample\_@aup\_p\_allianianut@\_Word
```

Figure 7: Quering the individuals in the English, Inuktitut, and Inuktitut Romanized aboxes using RacerPro and the JRacer API

4.2.2 Contrastive Features

In addition, if a certain muscle is not used in a language to make a contrastive difference (two words have different meanings) then this muscle is not specified for all speech sounds and can serve to augment the distinctions already available. For example vibration of the vocal cords are not used to distinguish sounds in Korean or Inuktitut.

In addition whether or not a muscle (or phonological feature) is contrastive can depend on the features's position in an utterance. In German voicing is not contrastive at the end of a word, in English voicing is not contrastive at the end of a phonogical phrase. Phoneticians/phonogists gain experience in contrastive features and feature changing processes in different positions of a word. These changes are important in-order to uncover the underlying morpheme and reduce vocabulary size of morphemes.

5 Discussion

Coming from a background of linguistics, with some prior understanding of Semantic formalisms, the Semantic Web formalism, and conceptualization, implementation and debugging process really forced me to think about how we can *make use* of Semantics, how we can approximate human reasoning in a way that is implementable, and executable within a reasonable wait period. I was forced to think about what was underlying my project, how can I capture the semi-statistical ideas behind natural human reasoning of a “prototypical” letter of Inuktitut, vs Spanish? How can I capture the phonotactics of Spanish as compared to English (which allows lots of consonant clusters)? At what level of zoom should I place the individuals in my model?

Over the course of development I asked myself whether an Ontology was the right tool to do Language Identification of a Text vs. a Word. My answer is that an ontology is almost the only way to provide a reasonable guess as to a word's langauge source, if only one word is availible, but a simple Perl/Python script could provide a reasonable guess for text (larger than 20 words, with a training set of roughly 1000 words).

Although the task of deciding the language of a text is more important than that of deciding the language of an isolated word, this can be important in multilingual documents, such as those produced in Canada or in Quebec where official documents must be bilingual and often alternate paragraph by paragraph. A purely statistical model would suffer from high innacuracy on this task, while a concept driven Ontology such as Phonont would fair better. This task is also very usefull in Spell checkers and grammar checkers. Often spell checkers opperate on word edit distance. The connections between letters is made explicit by Phonont, so that if somone types “sentance” rather than "sentence" a plugin to Phonont could provide a pop up bubble suggesting a "similar vowel" as English speakers pronounce unstressed vowels the same, all unstressed vowels would be offered. While spell checkers are already being used and have tools which deal rather well with existing vocabulary items the tools do not work well with novel word such as "systematicities." However, Phonont can make intellegent suggestions to the users on how, if that were a word, it would be spelled. These are tools for native speakers of a language, but there are more important (more necessary and more profitable) tools for non-native speakers, especially in the area of Grammar checking. English native speakers are better at checking their own grammar than any machine could ever be. However, phonont can help for typos made by francophones (using "hate" for "ate", "here" for "ear"') by using the similarity relation: /h/, a glottal fricative is one feature different from a glottal stop, which appears before all vowel initial words. This is the reasons that humans (francophone users of English) confuse these words, and this reason is encoded in Phonont.

Apart from usefull tools in text processing Phonont, was intended to improve morphological segmentation in languages which write allophonically rather than phonemically, such as Turkish which has vowel harmony endings -ler, -lar which sould be segmented off from stems. The risk of not having a processing stage which identifies if a language has vowel harmony (for unsupervised segmetnation on undefined languages) is that the -l might be segmetned into the root, essentailly doubling the vocablary size of all nouns in the languagage (-ler/-lar) is the plural suffix, and as such will apear on most nouns in the corpus). This goal can be tested by sending queries (about concepts, rather than individuals) out of a morphological analyser. My morphological analyzer which I will work on in my masters is written in Perl and bash scripts so it requires a porting of Phonont to a Unix environment for testing, this is what I will do durring Christmas break.

References

[1] Antoniou, Grigoris and Van Harmelen, Frank. 2008. *A semantic web primer.* Second edition. Cambridge, MA: The MIT Press.

[2] Brent, Michael R and Xiaopeng Tao. 2001. "Chinese text segmentation with mbdp-1: Making the most of training corpora." In *39th Annual Meeting of the ACL,* pages 8289.

[3] Calvanese, D. et al. 2007. "Software Tools for Ontology Access, Processing, and Usage." *Deliverable TONES-D21*. Thinking ONtologiES. www.tonesproject.org

[4] Creutz, Mathias and Krista Lagus. 2005. "Inducing the Morphological Lexicon of a Natural Language from Unannotated Text." In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05),* pages 106-113, Espoo, June.

[5] deMarken) de Marcken, Carl. 1995. "Acquiring a lexicon from unsegmented speech." In *33rd Annual Meeting of the ACL,* pages 311313.

[6] Kamholz, David. 2006. "An Ontology for Sounds and Sound Patterns." Max Planck Institute For Evolutionary Anthropology, Leipzig. http://emeld.org/workshop/2005/papers/kamholz-paper.html

[7] Konstantinou, N. et al. 2008. "Ontology and database mapping: a survey of current implementations and future directions." *Journal of Web Engineering*, Vol. 7, No.1 (001-024).

[8] Link, Godehard. 1983. "The Logical Analysis of Plurals and Mass Terms : A Lattice-theoretical Approach." In *Meaning, Use and Interpretation of Language,* R. Bauerle, C. Schwarze and A. von Stechow eds. Gruyter, New York.

[9] Protege Wiki http://protegewiki.stanford.edu/

[10] RacerPro Reference Manual 1.9.2 BETA http://www.racer-systems.com/

[11] RacerPro Users Guide 1.9.2 BETA http://www.racer-systems.com/

[12] Sene, S. and A. Shirke. 2009. "Generating OWL ontologies from a relational databases for semantic web." *International Conference on Advances in Computing, Communication and Control (ICA C3'09)*

[13] Stoilos, Giorgos et al. 2005. "Fuzzy OWL: Uncertainty and the Semantic Web." *Proc. Intl Workshop OWL: Experiences and Directions.* http://image.ntua.gr/papers/398.pdf

[14] Tho, Quan Thanh, et al. 2006. "Automatic Fuzzy Ontology Generation." IEEE Transactions in Knowledge and Data Engineering, Vol.18, No.6.