Master Thesis

Basic level detection: Learning from corpus characteristics and synthetic features

Author: Haochen Wang (VU: 2698251 UvA: 13500198)

Internal supervisor:	Adam S.Z. Belloum
External supervisor:	Laura Hollink (HCDA, CWI)
2nd reader:	WHO

A thesis submitted in fulfillment of the requirements for the joint UvA-VU Master of Science degree in Computer Science

June 29, 2022

"I am the master of my fate, I am the captain of my soul"
from Invictus*, by William Ernest Henley*

Abstract

Context. at the end

Goal. at the end

Method. at the end

Results. at the end

Conclusions. at the end

Contents

CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

This section includes some motivations behind the work, explicitly or implicitly highlights the research question, provides a high-level explanation of the solution, and describes the contributions.

2

Background

2.1 Literature Study

This section provides the necessary context of basic level categories theory, Random Forest with SMOTE algorithm, Word2Vec, and BART to help readers understand.

2.1.1 Basic Level Categories Theory

Basic level as a level of abstraction in taxonomy is observed in 1958 by psychologist Roger Brown. He stated a phenomenon that there is a preferred level of names that is the most useful in most contexts(5). However, he did not give the level nor names at this level a definitive term or description. A formal name for basic level categories and a systematic theory of basic level categories are developed by psychologist Eleanor Rosch in 1976(6). Research on the basic level categories has been across diverse disciplines. Studies in psychology, anthropology, linguistics, and library and information science have more or less covered the theory of basic level categories to measure perception, communication, and behavior(7).

Besides Brown and Rosch, linguist George Lakoff raised a research question what do categories of language and thought to reveal about the human mind. He demonstrates that basic level categories are 'human-sized' and depend upon human interactions with objects in a category(8). Observed in the field of library and information science, concepts in basic level categories have been demonstrated to have more possibilities to be shared across classificatory systems than others by Rebecca Green(9).

Although there are many publications on basic level categories, they hardly give a specific mathematical definition to the basic level categories. Most of the related work describes the basic level categories theory with an intuitive idea, such as categories containing the

most information or people would react fast to these categories. To clarify what the basic level categories mean to humans, a definition of basic level categories in this project is stated from aspects of both semantics and quantified method. The semantic one is cognitive economy which explains that people tend to make less effort to understand and react to the basic level categories. The quantified one is cue validity(6) which describes how much information is gained by people from the basic level categories. In this way, basic level categories, a terminology of psychology, can not only be described by a level of abstraction, but measured by a mathematical formula as well. The cognitive economy and the cue validity are essentially the same and used to define the basic level categories.

Some terminologies about basic level categories are similar. The basic level is a level at which concepts belong to basic level categories. Therefore, basic level and basic level categories have the same meaning in this paper.

2.1.1.1 Cognitive Economy

Humans perceive the real world with correlational structures. The perception is from cognitive processes which tend to minimize exertion and resources cost during processing. Further, cognitive economy concerns the relevance and simplicity of a categorization scheme and knowledge representation(10). From the aspect of cognitive economy, the basic level, as a criterion of binary classification, can result from a combination of the following two principles(6).

1. **Predictable property.** Concepts in the basic level would have many predictable properties which can be known from each other or any one of them. The attribute of predictable properties leads to forming a large number of categories. There are discriminations in each category and one of them holds the fine enough differences to distinguish each concept belonging to it. This category is possibly to be the basic level.

2. **Relevant differentiation.** The categorization scheme aims to reduce infinite differences to an appropriate degree of proportions among concepts within one category. The appropriate degree of proportions behaviorally and cognitively depends on the purposes. Otherwise, it would not differentiate a concept from others unless the differentiation is relevant enough for the purposes.

The two principles look contradictory in how humans perceive the world with categorization. They emphasize an appropriate degree of differentiation according to human'

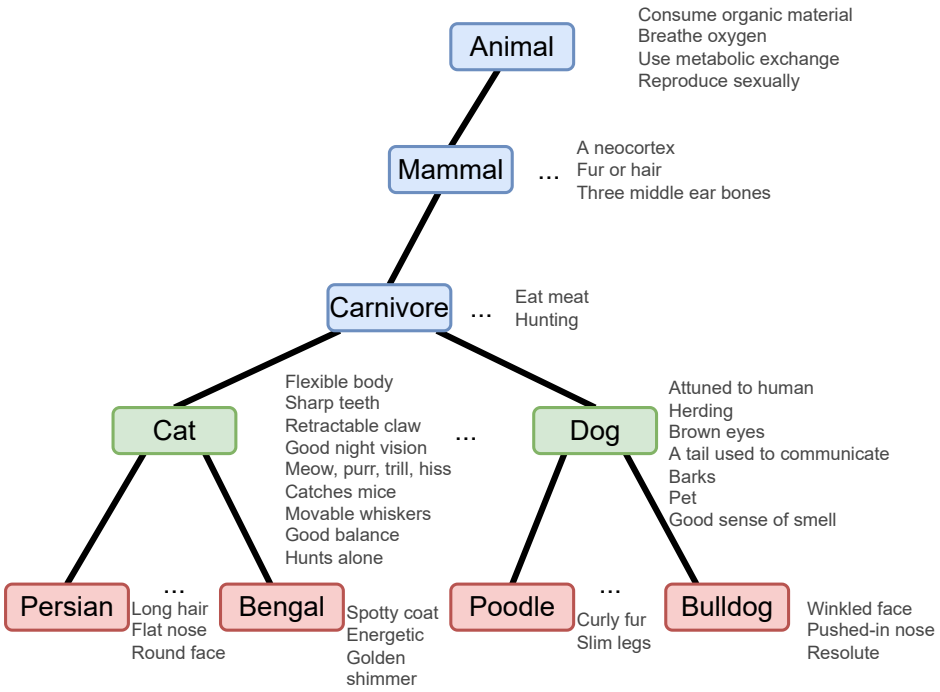

Figure 2.1: Concept hierarchy with properties example (1)

interaction with the world under miscellaneous situations. For example, assuming a concept hierarchy(1) in Figure 2.1, the concept of cat and the concept of dog can be in the basic level according to the principles of cognitive economy. In the hierarchy, concepts of upper cat and dog are more abstract whose properties can not be the same predictable as cat and dog. Conversely, concepts lower than them are too specific whose properties indicate only slight differentiation. The concept of cat and the concept of dog both hold as many predictable properties as possible to distinguish them from the others (Predictable property). Meanwhile, a category of the concept of cat and the concept of dog can be the most appropriate proportions of differentiation for a general perception by humans (Relevant differentiation).

By basic level categories, concepts more abstract or general than those in basic level are superordinate concepts, i.e. hypernyms in a hierarchy. Concepts more specific than or below the basic level are subordinate, i.e. hyponyms in a hierarchy. With the basic level theory, humans can sketch the real-world correlational structures.

2.1.1.2 Cue Validity

Cue validity, $val(cue)$, is based on conditional probabilities which typically include $P(BL|cue)$ and $P(\overline{BL}|cue)$ terms. $P(BL|cue)$ is the probability of a concept is **IN** basic level given the cue, while $P(\overline{BL}|cue)$ is the probability of a concept is **NOT IN** basic level given the cue. Qualitatively, $val(cue)$ goes up when $P(BL|cue)$ increases and(or) $P(\overline{BL}|cue)$ decreases. However, the strict mathematical form to calculate the cue validity is various. BEACH proposed probabilistic cues (Equation 2.1) to make inferences about objects' category (11). Here, the object can be seen the same as the concept in this paper.

$$E(k) = \frac{\sum_{d=1}^{n} P(k|d)}{n} \tag{2.1}$$

Where k is a specific category that one object probably belongs to. $E(k)$ can be regarded as a possibility of one object correctly expected to be in a given category. d is a dimension on which one object's cue is known. n is the number of the dimensions. $P(k|d)$ is the relative frequency with which one object's cue on each cue dimension (11). He improved the inference method in another paper. Another formula (Equation 2.2) was put forwards to recognize, assimilate, and identify a category for an object (12).

$$E(c) = \sum_{d=1}^{n} \frac{P(c|k,d)}{n} \tag{2.2}$$

Where c is a cue value under consideration as one object's unknown cue. $E(c)$ is the total evidence from the unknown cue dimension. $P(c|k,d)$ is a probability that a cue value c on an unknown cue dimension d is the best bet for the inference give one object's category k (12). Based on Beach's study, Reed updated the algorithm to calculate the cue validity and proposed a similar formula, Equation 2.3, which measures the cue validity by considering the frequency and the proportion of cues in categories (13).

$$CV(Category\_k, X) = \sum_{m=1}^{d} \frac{P(Category\_k|x_m)}{d} \tag{2.3}$$

Where k is an ordinal number of categories while m is an ordinal number of cues of X. $CV(Category\_k, X)$ is the cue validity value of concept X. $P(Category\_k|x_m)$ is the prior probability by $P = 1/(1+F)$, where F is the frequency with which the cue appears in the category (13).

According to the definitions of basic level, a concept with a larger cue validity can be more differentiated than others with a lower one. It is reasonable that the superordinate concepts have fewer attributes in common to have lower cue validities. Meanwhile, the

subordinate concepts share so many attributes among siblings that lead to lower cue validities. Concepts in the basic level maximize cue validity, in other words, these concepts reflect the correlational structure of the real-world environment best and are identified fast for humans.

When computing cue validity of a concept, the practical implementation does not directly conduct the Equation 4.2 because it contains a posterior probability that is impossible to count and calculate for training and testing. The detail of implementing cue validity of concepts will be discussed in Section 4.4.2.

2.1.2 Random Forest with SMOTE

Basic level detection is to categorize a concept into basic level or non-basic level which is a binary classification task in Machine Learning. Random Forest(14) with Synthetic Minority Over-sampling Technique(SMOTE)(15) will be used as a classifier to learn from synthetic features and to predict concepts whether are in the basic level or not.

Random Forest is an extensive version of Bagging, which is the abbreviation of Bootstrap Aggregating. It uses Decision Tree as a base learner and builds a Bagging aggregation. Furthermore, Random Forest introduces the random choice of attributes in the process of training. The core concept of the fundamental method, Bagging, is sampling and training for every subset of the attributes. They can be sampled for training a Decision Tree with a certain number of items. The training leads to a base learner. Hence, several base learners can be integrated or aggregated into a final Random Forest learning model.

Specifically, training a Random Forest classifier includes sampling attributes and choosing an attribute. Firstly, a subset of the total d attributes is sampled for each current node in the base Decision Tree in a bootstrap way. The size of every subset is k. Secondly, the most optimal attributes from the subsets are chosen to generate their child nodes respectively. In this way, the parameter k controls the degree of randomness introduced. In general, according to (14), the recommended value would be:

$$k = \log_2 d \tag{2.4}$$

Random Forest is relatively simple to implement while needs low computational cost. Moreover, it has performed powerful abilities and performance in many Machine Learning tasks. Besides only an initial bootstrapping on a training set, Random Forest exploits bootstrap to attributes. Both the self-sample perturbation and the self-attribute perturbation enable better performance of generalization via increasing the bias of individual

base learners. Therefore, Random Forest usually converges to lower generalization errors with the increment of the number of base learners.

When training a Random Forest with imbalanced datasets, SMOTE algorithm for sampling can achieve better classification performance. In general, SMOTE is a method that oversampling minority classes and undersampling majority classes(15).

For undersampling the majority classes, the samples are removed at random until the percentage of samples in the majority classes and the minority classes reaches a specified value. For oversampling the minority classes, besides taking minority samples, it creates synthetic examples of each minority instead of directly replicating with replacement. The synthetic examples are calculated with the K-Nearest Neighbor algorithm. Discovering the nearest neighbors of a minority sample, each difference between the minority and its nearest neighbors is multiplied by a random number ranging from 0 to 1. Then, the new examples can be added to the dataset for training which leads a classifier to have greater decision regions but less specific than without such oversampling.

2.1.3 Word Embedding: Word2vec

Word embedding is an important technique in Natural Language Processing that words are mapped to vectors of real numbers. It is a necessary procedure in modeling a language and learning features from textual data to numerical representations. Word embedding aims to capture the meaning of a word in semantic similarity, syntactic similarity, and relations with other words which makes natural language read and processed by computers. A well-trained set of word embeddings will place similar words close to each other in the vector space. Based on the real-number representation of words, further computation and algorithms can be implemented.

Word2vec(16) is one of the most popular techniques to learn word embeddings using multi-layer recurrent neural networks. There are two main training algorithms for Word2vec, the continuous bag-of-words(CBOW) model and the skip-gram model. The major difference between the two models is that CBOW uses context to predict a target word while skip-gram uses a word to predict the target context. According to Mikolov(17), the skip-gram model is more suitable for representing not frequent words. To hit lemmas of the concepts in the dataset Section 3.2 as many as possible, a Word2vec-based repository named *ConceptNet Numberbatch*[1] can provide finely pre-computed word embeddings trained with data from *ConceptNet*(18) using the skip-gram model. Compared to other

[1] `https://github.com/commonsense/conceptnet-numberbatch`

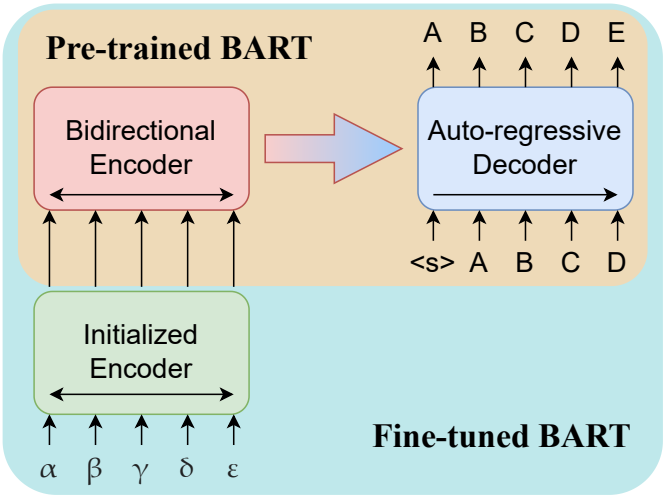

Figure 2.2: Architecture of BART(2)

pre-computed word embeddings, *ConceptNet Numberbatch* is able to cover most of the lemmas in our dataset which guarantees to eliminate missing vectors as few as possible. The vector representation with the semantics of a lemma of concepts in a hierarchy can be looked up in the dictionary of *ConceptNet Numberbatch*.

2.1.4 BART

BART, Bidirectional and Auto-Regressive Transformers, is a sequence-to-sequence denoising autoencoder model(2). It is one of the effective language models for text generation and comprehension tasks, such as machine translation. By fine-tuning BART, an end-to-end model can be trained which can learn a mapping from source English words to their semantic features. In this paper, BART will be used as a pre-trained model for tokenization and a fine-tuned autoencoder for semantic feature generation.

The architecture of BART follows the standard Transformer(19). It is implemented with a bidirectional encoder and an auto-regressive decoder,shown in the yellow area of Figure 2.2. The pre-trained BART is denoising because the input training data is corrupted text with masks(attention) and the goal is to reconstruct the text which is noticed by the masks. Both the encoder stacks and decoder stacks contain 12 identical layers(19). The encoder of the pre-trained BART then can be used as a tokenizer for English words. It gives a vector of identity with an attention mask to each word or phrase which implicitly represents the meaning.

After fine-tuning the BART with an additional encoder, named Initialized Encoder, the

new model is designed for machine translation tasks, shown in the green area of Figure 2.2. The pre-trained BART without the embedding layer is used as a decoder. The new encoder is trained to map the input source text into an intermediate representation which can be denoised by the pre-trained BART (2). The BART will be fine-tuned with English semantic feature data, to be introduced in Section 3.5. Using the fine-tuned BART, the end-to-end model of translation can help to generate semantic features from a word. In other words, the translation is a mapping from one concept to its semantic features. The pipeline and detailed process of fine-tuning will be discussed in Section 4.4.2.

2.2 Related Work

Describe here scientific papers similar to your experiment, both in terms of goal and methodology. Two paragraphs for each paper (we expect about 5-8 papers to be discussed). Each paragraph contains: (i) a brief description of the related paper and (ii) a black-on-white description about how your work differs from the related paper. You may place this section immediately after the Background section, if necessary.

2.2.1 Rule-based Heuristics

Mills et al.(20) built a rule-based system with heuristics to identify basic level categories automatically. Their approach is to evaluate a cumulative set of rules defined by themselves. The system constrains concepts being the basic level with some boundaries of the rules. Initially, there are 52 rules in two types: filtering rules and voting rules. They used several resources of corpora, dictionaries, and toolkit to formulate the rules. After experiments of training and developing, there are 8 chosen filtering rules with parameters and 4 selected voting rules left for relaxation using a greedy search scheme.

Although the system can identify the basic level with a relatively high accuracy of 77.0% and classify automatically, the data gathered was limited, 194 categories in total. For the reason that some categories do not have corresponding synsets in WordNet, the categories used in the experiment are even fewer, 152. Moreover, there could be many important features ignored because of the removal of weak rules. It might not work well with concepts outside the 152 categories because the rule-based system is trained and developed with only 100 categories. In this thesis, experiments are conducted with more annotated concepts, up to 839. and have different models designed to guarantee the generalization of the method for predicting the basic level.

2.2.2 Machine Learning-based Classification

Recently, more related research of predicting the basic level focuses on Machine Learning. Concepts can be categorized into the basic level or others using several kinds of classifiers. With Machine Learning algorithms, predicting the basic level is regarded as a classification task. Moreover, appropriate feature engineering can improve the accuracy and efficiency of the predicting.

Hollink et al.(21) aim to predict whether concepts are the basic level in a concept hierarchy. They trained five kinds of classifiers from three types of features: lexical features, structural features, and frequency features. The classifiers are trained by Latent Dirichlet Allocation(LDA), Decision Tree, K-Nearest Neighbors, Support Vector Machine(SVM), and Random Forest. The lexical and structural features are extracted from WordNet(3), while the frequency is from Google Books Ngram(22). They present a method to classify concepts from a conceptual hierarchy into a basic level and non-basic level using Random Forest. The models are trained in the setting of within one domain and across domains. The local model, whose training data is within a domain, results in the best performance under three domains. They argue that concepts that are difficult to label for humans are also harder to classify automatically.

The method Hollink et al. considered and the features they chosen only concern the structure of concepts in a hierarchy and their lemmas morphology. The lexical features and structural features do implicitly contain some semantic relations among synsets from their hypernyms and hyponyms. The implicit semantic relations could indicate the subordinate relationship, however, might not be able to summarize meanings of one concept(synset). In this thesis, semantics of concepts is explicitly represented by their cues generated by the fine-tuned BART. The cues of a concept could explain the synset directly rather than inferred from the subordinate relationship. The method proposed in the thesis adopts both of the implicit semantics from the hierarchy and the explicit semantic features by cues.

Henry(23) focuses on the features from corpora. She raises a research question that what corpora properties are useful in predicting the basic level. It is through learning the basic level with varying corpora of different discourse types, audience ages, and sizes in words. She concludes that larger corpus sizes have more reliable results. And comparing smaller samples of the same size, those containing spoken discourse and discourse directed at children provided more reliable results than written text aimed at a general audience. The features from child spoken corpus can be important indicators to learn and detect the basic level.

It reveals the significance of the type and the size of frequency resources. However, the aggregations of frequency features from different corpora are not the same. The performance of accuracy for predicting the basic level is not improved significantly from them. Henry did not consider semantic features of concepts either.

Chen and Teufel(1) present the first method for the detection of the basic level at scale using Roach-style semantic features which contain cue validity, according to their statement. They adopt three methods of generating semantic features for synsets in WordNet: textual features from Wikipedia pages, Distributional Memory(24), and BART. The languages are English and Mandarin. The synthetic textual features include structural features, lexical features, Word2Vec, frequency features, cue validity, basic level page rank, and semantic features. Support Vector Machine is used to train the classifier.

Although Chen and Teufel find that BART is capable of generating indicators to improve the detection of the basic level, they did not clarify the mechanism of BART nor the functionality of the generation. The best model in their experiments performs 75.0% accuracy of English basic level detection and 80.7% in Mandarin on their test set. However, the dataset only contains 433 concepts which is carefully selected and not directly from a developed hierarchy.

2.2.3 Context-aware in Folksonomies

Chen et al.(25) put forward an algorithm to detect the basic level among various contexts from folksonomies(26). The folksonomies contain implicit semantics from creating and managing tags in web resources annotated by users. They model instances, concepts, and context in the folksonomies for mining semantics. Contextual category utility, inspired from category utility(27), is proposed to predict the basic level. The modeled concepts are detected as the basic level when they have the greatest value of the contextual category utility.

Chen et al. though considered semantics when predicting the basic level and under large-scale web resources. The concepts are discovered from the web which are not in a hierarchy. The results depend on the contents and quality of the resources. Their method is not appropriate for predicting in a hierarchy because the folksonomies would miss many concepts from synsets. They neither use indicators of lexical, frequency nor structural characteristics.

3

Data

In this section, the data used and methods to acquire are explained.

3.1 WordNet

Princeton WordNet[1] is a lexical database for English which organizes sets of synonyms representing lexicalized concepts (3). The synonymy is from a semantic relation of synonymy which is the basic principle to arrange concepts. The sets of synonyms, known as *synsets*, are used to stand for word senses that is regarded as concepts in this paper. The canonical form or morphological form of a word from the synonyms is one of the *lemmas* of the synset. The meaning of each synset is named sense. By these definitions, the lexical semantics can be described in terms of the relations between their senses. There are over $166,000$ relations, which are represented in pairs of a lemma and a sense, and more than $117,000$ *synsets* in WordNet (3).

Another important semantic relations are hyponymy and hypernymy which are the transitive relations between *synsets*. *Hyponyms* and *hypernym* can shape definitive paths from the superordinate to several Subordinate. The paths hold semantics and each synset for nouns usually has one *hypernym*. Therefore, concepts in WordNet are organized in a hierarchical structure of the lexicon. Further, every synset with its *hyponyms* and the relations can be seen as a hierarchy of lexical knowledge. One hierarchy of the concepts in the annotation dataset is shown in Figure 3.1. The synsets in light red are the domains in the dataset Section 3.2. Synsets in yellow are the hyponyms of the domains and synsets in blue are their hypernyms. The root of the hierarchy is the synset of *entity.n.01*.

[1] `https://wordnet.princeton.edu/`

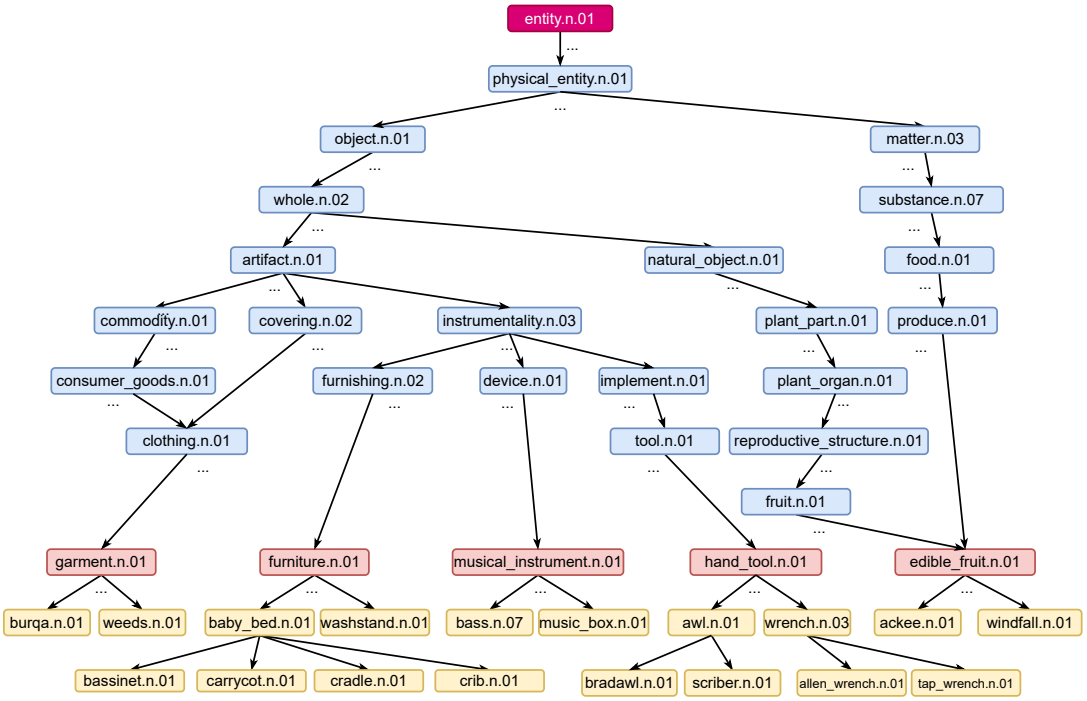

Figure 3.1: Hierarchy of concepts in WordNet (3)

As mentioned in Section 3.2, concepts to be predict can be the *synsets* in WordNet. Moreover, the methodology proposed can be executed to detect the basic level with all the concepts in *entity.n.01*. WordNet database and its API can be accessed by NLTK[1] WordNet Interface[2].

3.2 Basic Level Annotations

The dataset where concepts are labeled with basic level or non-basic level is inherited from Hollink et al. (21) and Henry's research. There are three domains from Hollink's dataset and two domains from Henry's. The domains are *hand_ tool.n.01*, *edible_ fruit.n.01*, *musical_ instrument.n.01*, *furniture.n.01*, and *garment.n.01* in WordNet. The labeled dataset is called gold standard. Originally, the gold standard labels concepts in the basic level, or the superordinate or the subordinate of the basic level. In this paper, superordinates and subordinates are merged into a class of non-basic level.

Concepts in the gold standard are labeled manually by three annotators who are provided with an annotation protocol. The protocol includes instructions of this labeling task, descriptions of the basic level, characteristics of the basic level, and how to find the basic level in the hierarchy of WordNet. The most important part is a checklist helping label the basic level. In addition to the checklist, the annotators may access necessary information from Wikipedia and Google Search Engine. Using the annotation protocol, concepts labeled as the basic level can be as close as possible to the Roach's definition of the basic level, discussed in Section 2.1.1.

After processing of the gold standard, the dataset to be used in experiments is summarized in Table 3.1. It shows distributions of the number of concepts in each domain. The data is imbalanced that concepts at the non-basic level are $2 \sim 8$ times more than those at the basic level. Considering the definition of the basic level, it is reasonable that the basic level is less but contain more information in one domain. The settings of the training set, validation set, and testing set will be discussed in Section 4.1.

3.3 Textual Corpora

To answer the first research question about frequency features, different resources for calculating frequencies of concepts should be considered. The aim is to gather the frequency

[1] NLTK: Natural Language Toolkit

[2] `https://www.nltk.org/howto/wordnet.html`

Domain	Basic level	Non-basic level	Total
hand tool	25	108	133
edible fruit	57	99	156
musical instrument	47	79	126
furniture	20	163	183
garment	26	215	241
Total	175	664	839

Table 3.1: Summary of Basic Level Annotation Dataset

of lemmas in each concept from different discourse types, different target audiences, and in various sizes of resources. Therefore, four corpora with different characteristics are extracted to be used as the frequency resources. Summarized in Table 3.2, they are the KBNC, the CABank English corpus(CABNC) (28), the CHILDES (29), and the British National Corpus(BNC) (30),

Text BNC is a British English corpus which contains around 100 million words from a wide range of written and spoken resources. It records abundant British English from the late 20th century and is released in 2007 as *BNC XML Edition*[1]. Approximate 88 million words of written records are extracted and marked as BNC Written corpus for the frequency feature under a general written corpus. Meanwhile, there are around 1 million records of them specific for children. They are wrapped as KBNC which is a children written corpus.

CABNC is built by re-transcribing naturalistic conversations from Audio BNC, a subcorpus of BNC which originally contains about 7.5 million words in a type of audio. Albert et al. converted the transcripts into CHAT files (31) and made them public open-licensed[2]. CABNC initially has around 4.2 million words. However, from the latest version released only 2.4 million words can be parsed from CHAT files by *PyLangAcq*[3]. The parsed words compose CABNC for calculating frequencies of concepts under a general spoken corpus.

CHILDES is one of the components in the TalkBank system specific for child languages[4]. 16 corpora from British English consist of a new corpus, simply named CHILDES. The new CHILDES contains around 5.7 million words that are transcribed from conversations,

[1] `http://www.natcorp.ox.ac.uk/`
[2] `https://ca.talkbank.org/access/CABNC.html`
[3] `https://pylangacq.org/`
[4] `https://childes.talkbank.org/`

Corpus	Discourse Type	Target Audience	Size Approx.	Description
KBNC	Written	Children	1 million	Subset of the BNC specific for children target audience
CABNC	Spoken	General	2.4 million	Re-transcribed from a subcorpus of the BNC
CHILDES	Spoken	Children	5.7 million	Composed of 16 subcorpora
BNC Written	Written	General	88 million	British English in the late 20th century

Table 3.2: Summary of Corpora for frequency features

audios, or videos. It will be used to extract frequency features within a children spoken corpus.

3.4 Google Books Ngram Corpus

To have a larger corpus for extracting the frequency features, Google Books Ngram Corpus(Google Ngram) (32) can be another resource which is a written corpus for general audiences. It is an enormous analytical repository of printed publications. Similar to Hollink et al. and Henry's study, frequency features from Google Ngram could be a set of important indicators for the classification. The corpus has three versions. This paper adopts the third version released in 2020. It contains millions of books published since 1500s. Although the accuracy number of tokens in the Google Ngram version 3 is not documented, it can be sure that the amount is larger than the second version, which is over 468 billion tokens (22). And it was updated by billions of records annually during 2012 to 2019. The ngram data used is all the entries in Google Ngram Version 3 from 1500 to 2019.

Google provides a web-based service to search the frequencies of words by years on Google Books Ngram Viewer[1]. For every concept in the dataset, lemmas of the synset are listed by the NLTK WordNet library correspondingly. It can include all the words within

[1] `https://books.google.com/ngrams`

the concept so that the frequency of a concept is more complete to represent its feature. Unfortunately, there is no official API for querying frequencies in a large-scale productive mode. The frequency data has to be obtained by a web crawl that posts requests for the frequency of a lemma and gets its response. The response can be parsed and analyzed to have valid frequency data. The returned data contains frequencies of a word(lemma) in the given period of years.

However, it is found that continuous requests to the Google Ngram Viewer would trigger an exception of *503 Service Unavailable* and respond the null data. The reason is that Google set a limitation of request times to protect its server and services. The policy of the Google server request limit is discovered to be likely 75 requests every 560 seconds. To solve this challenge, some implementations set sleep time between every request, but it could cost over 4 hours for querying all the dataset. According to the policy, an optimal crawler is implemented to speed up the procedure of the querying. It reschedules the sleep strategy to 72 requests then wait 580 seconds every round instead of sleeping 10 seconds between each request. With the new strategy, the time of the querying reduces to 3 hours for all the concepts. Besides, the optimal crawler is encapsulated as a Python class which can automatically query the frequencies given a concept and a period of years. Moreover, it provides an option that can aggregate the frequencies of a concept in a range for years into the maximum, minimal, mean, and standard deviation. The aggregations may help feature engineering to be discussed in Section 4.3.2.

3.5 English Semantic Feature Database

Mentioned in Section 2.1.4, BART will be fine-tuned with semantic features of words. A project of producing English semantic features[1] provides a database of 4436 concepts with their semantic features by Buchanan et al. (4). The database is organized with word pairs $(concept, feature)$ which represent the close relation of their meanings and other statistics on semantics. They built the database by examining the answers of concepts obtained from crowdsourcing and processing their feature frequencies respectively.

The entire database has 69284 records of the word pairs with the part of speech (POS) as well as the statistics on features and frequencies. The features have been 'translated' to lemmas (lemmatization) using Snowball stemmer (33). Only the pairs of lemmas are adopted in fine-tuning the BART. The detail of the English semantic feature database is summarized in Table 3.3.

[1] `https://wordnorms.com/`

POS	#Concepts	#Records	#Lemma Pairs
Noun	3125	51923	32051
Adjective	663	7511	3929
Verb	548	8772	6045
Other	100	1078	591
Total	4436	69284	42616

Table 3.3: Summary of the English Semantic Feature Database

4

Method

This section presents the method to deal with features and predict basic level.

4.1 Classifier

Concepts are categorized into ~~the~~ basic level or ~~the~~ non-basic level. The classification task in this thesis is ~~finished~~ by a machine learning classifier, Random Forest with SMOTE algorithm. It ~~will be~~ used to measure the performance of the synthetic features.

There are several reasons to adopt Random Forest with SMOTE algorithm. The classifier ~~developed~~ in both Hollink et al. (21) and Henry (23) ~~is Random Forest with SMOTE~~. ~~It has been turned out to be the~~ best classifiers for the basic level in Hollink et al. (21) and reused by Henry. Performance improved by the features in this thesis is easier to be compared with others' using the same classifier. Another reason is ~~from~~ advantages of Random Forest itself. It introduces randomization that helps ~~avoid~~ over-fitting, ~~in the meanwhile~~, it can be trained fast and efficiently even with large-scaled data. The input features can be both discrete and continuous variables without normalization. Moreover, after training and validation, it can return the importance of each feature which helps to analyze the effeteness of the features. ~~Therefore, Random Forest with SMOTE algorithm will be used as a benchmark classifier~~.

The Random Forest has 1400 Decision Trees as base learners which are trained with sub-dataset sampled with replacement from the dataset. Because of the method of bootstrap to build up the Random Forest, out-of-bag samples are feasible to estimate the generalization score of the classifier. For each Decision Tree in the Random Forest, Gini impurity is used to measure the quality of a split with a node. The maximum depth of each tree is set to 50 which can control over-fitting and make the training fast. It is required that each split

leads to at least two child nodes and each node has at least one instance from the training data.

We also try Support Vector Machine(SVM) ~~a~~ classifier ~~for the reason that~~ some semantic features are made up of vector-based embeddings. The SVM is trained with Radial Basis Function kernel, $\exp(-\gamma||x - x'||^2)$, where x and x' are both embedding vectors. After tuning by grid search, the best setting of hyper-parameter γ is scaled by $\gamma = 1/(n * var)$, where n is the number of the vector dimension and var is the variance of the input matrix.

The SVM is only used for classifying with the semantic feature of word embeddings in Section 4.4.1. The Random Forest as the benchmark of the method is the main classifier used to learn and predict the basic level.

4.2 Structural Feature Extraction

Structural features include lexical features of concepts and relational features of them extracted from WordNet. The synonymy, hypernymy, and hyponymy of a concept in WordNet convey semantic relations which reflect senses of the concept with its superordinates and subordinates. According to the cognitive economy in Rosch et al. (6), the relational features would be important indicators to classify whether a concept is in the basic level that carries the most information and costs minimal efforts. Moreover, discussed in Section 2.1.1.1, the relational features in the hierarchy, WordNet, naturally represent a correlational structure of the real-world knowledge which is significant in the basic level theory.

The basic level can be learned also from the lexical features. As pointed out, one of the characteristics of the basic level concepts is that they are generally denoted by shorter and more polysemous words (34). Therefore, the character length of lemmas in a concept and the number of the lemma polysemies would be important features for predicting the basic level concepts.

Hollink et al. (21) and Henry (23) both considered these structural features. Their data of lexical features and relational features is referred in ~~this thesis~~. Only some of WordNet features in Henry's work will be selected and reused in our method. The following structural features are extracted and to be trained by the classifier.

- The number of the direct hypernyms of a concept
- The number of the total(direct and indirect) hyponyms of a concept
- The normalized number of part-whole relations related to a concept

- The normalized depth of a concept from the root synset
- The normalized character length in the gloss of a concept
- The shortest character length of lemmas in a concept
- The number of lemmas in a concept
- The maximal number of polysemies of lemmas in a concept

4.3 Concept Frequency

For the concept frequency feature, the sources of calculating the frequency of concepts are focused. To answer the first research question, corpora with different characteristics are firstly compared to extract the frequency features which contributes the most to the performance. Then, according to the most reliable corpus characteristics of predicting the basic level, frequencies of the concepts will be acquired (in Section 3.4) and processed by feature engineering.

4.3.1 Corpus Characteristics Comparison

Frequencies of concepts can be extracted from various corpora. Roach et al. found that ~~the~~ basic level concepts are prominently used in daily communications, specially in communication with children. She also argued that the basic level could be the earliest categories sorted and named by children (6). Therefore, concept frequencies extracted from spoken discourses and child target audience corpora might improve the performance of basic level prediction. The corpus characteristics compared ~~in the method~~ are the size of a corpus, the discourse type of a corpus, and the target audience of a corpus. The comparisons should take the size of the original corpus into account, discussed in Section 3.3.

Unlike Henry (23) adopted a range of statistics to calculate frequencies, the frequency of a concept in this method is purely defined by occurrences of its lemmas. The sum of occurrences of the lemmas in a corpus represents the frequency of the concept ~~in the corpus under its size~~. The number of the sum ~~will be~~ the frequency feature directly fed to the classifier.

The first comparison concentrates on the size of a corpus. The hypothesis is that performance would become better ~~with an increment of the size of the corpus~~. Based on the structural features, the frequency features ~~will be~~ verified respectively by the benchmark Random Forest. With the different types and sizes of corpora sampled, multiple classifiers

~~will be~~ trained and validated under the models and experiment settings designed in Section 5.1.

The second comparison focuses on the discourse type of a corpus. The hypothesis is that performance of the classifier trained by the frequency feature from a spoken corpus is better than that from a written corpus. This reflects to the previous research that the basic level concepts are likely to be mentioned in daily communications and be the most used in language (6). Intuitively, frequencies of the basic level in a spoken corpus would stand out and be greater than those in a written corpus. Under this assumption, the frequency feature from a spoken corpus is a more important and effective feature for the classifier. The comparison ~~will be~~ conducted with a series of Wilcoxon rank-sum tests.

The third comparison gives attention to the target audience of a corpus. The experiments in Rosch et al. (6) showed the basic level concepts are the first used by children developing language. According to this statement, the hypothesis is that performance by the frequency feature from a corpus specific to children is better than that from a general audience corpus. Similar to the second comparison, the performance of the benchmark classifier trained with the frequency feature from a child specific corpus and from a general audience corpus will be compared by Wilcoxon rank-sum tests.

It is worth noting that both the second and the third comparisons consider the size of the frequency source, which is regarded as a primary corpus characteristic in this method. The design of the experiments ~~will be~~ clarified in Section 5.2.

4.3.2 Frequency from Google Ngram

According to the results of the Wilcoxon rank-sum tests in Section **??**, it would be better to use a large, written, and ~~to~~ general audience corpus as the source of concept frequency. Google Book Ngram Corpus, ~~as far as I known,~~ is the largest corpus of printed publications available for public research. Therefore, the frequency features for predicting the basic level will be extracted from Google Ngram ~~in this method~~. Same as the decision on the frequency source by Hollink et al. (21) and Henry (23), they both selected Google Ngram to extract frequencies because frequency feature from it was the strongest individual feature among their experiments.

The frequency of a concept is the sum of frequencies of its lemmas in Google Ngram. By the data acquisition in Section 3.4, the frequency of a concept by year can be returned with an array whose elements represent the frequencies of the concept each year in Google Ngram. To discover whether the time period affects the performance, frequencies from Google Ngram corpus in the recent (based on 2019) 1 year, 5, 10, 20, 50, 100, 200, 400,

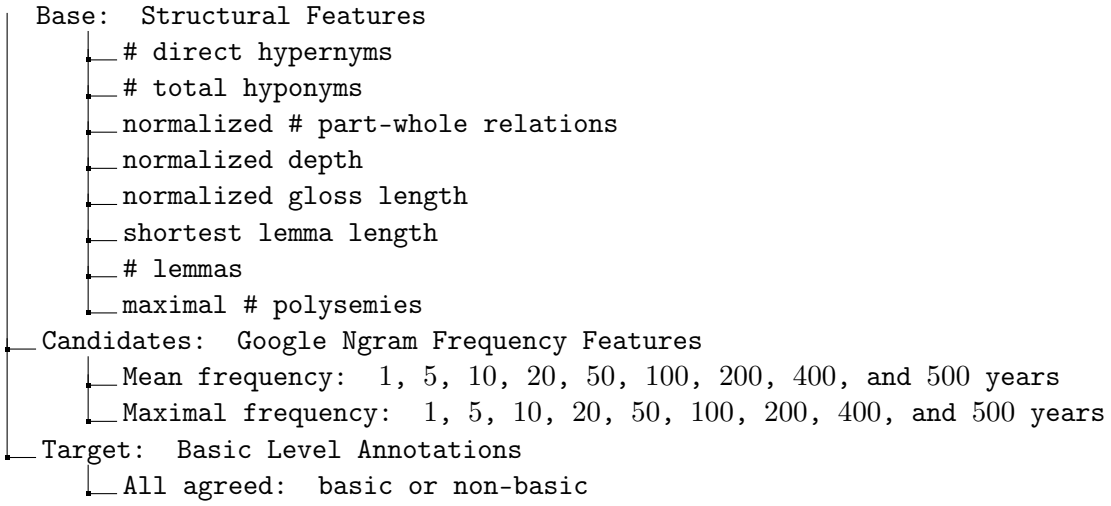

Figure 4.1: Frequency Feature Schema

and 500 years are gathered and stored for feature engineering. The frequencies in each array can be aggregated into the mean and maximal values as the features. The mean frequency would represent the average occurrences of a concept during a certain range of years while the maximal one would show how much was the most significant used in those years. After the processing, two groups of the frequency features of a concept each with 9 values are respectively the mean frequencies and the maximal frequencies among the 9 time periods.

The classifier keeps the same as corpus characteristics comparison in Section 4.3.1 except for the frequency feature selected. That is to say, the structural features remain as the base and pop each aggregated frequency feature into the training, shown in Figure 4.1. Hence, the performance of each Google Ngram frequency feature can be compare to discover whether there would be some patents related to the time periods.

To find out the best setting of the frequency features, a wrapper method of the feature selection is performed. Bottom-up and top-down approaches are deployed to check which combination would perform best with the metric of Cohen's kappa score (35). The importance of the features in the best setting from the benchmark Random Forest classifier are ranked and analysed in Section 6. The best combination of the frequency features will be added to the synthetic features together with the structural features and used to train the final classifier.

4.4 Generative Semantic Features

In the first place, "semantics" here does not describe nor deal with the general meaning of a concept. It neither needs to do so because some of the structural features in Section 4.2 surely have conveyed the general semantics from the lexical features or the relational features. "Semantic feature" in this method ~~especially~~ stands for the semantic representation of a concept which is key to models of semantic memory for facts (4) (36). To be concrete, semantic features ~~in this thesis~~ indicate the overlapping attributes of a concept defined by semantic similarity which can be regarded as cues in Section 2.1.1.2. For example, semantic features of a concept of *cat* might be *animal*, *pet*, *tail*, and *fur*. These features convey the most common and regular descriptions of a *cat*. Moreover, the semantic features might cover shapes, appearance, uses, gender, locations, characteristics, and etc. The aims of generating the semantic features are to measure the similarity of concepts and to create cues of concepts for predicting the basic level.

To learn the basic level from the semantic features, two methods are proposed to extract such semantics. ~~They are~~ word embeddings, and cues generated from BART. ~~There is a hypothesis for each method~~. ~~The first hypothesis is that~~ word embeddings, Word2Vec, would provide effective semantic features for predicting the basic level because it is able to measure a latent semantic distance between concepts in a hierarchy. ~~The other hypothesis is that~~ generating the semantic features from BART as cues would improve the accuracy of the prediction by using cue validity.

4.4.1 Word Embeddings

Word embeddings in this method are from *ConceptNet Numberbatch 19.08* trained by Word2vec, which is a task to represent words in the form of real-number vectors. The semantics of concepts is contained implicitly in the vectors. Intuitively, we use this model to compute the vector of a concepts as looking up in a dictionary. Each lemma of concepts are converted in to a 300-dimension vector. Unfortunately, there are 63 concepts in multi-word grams which are not directly in the vocabulary of the pre-trained *ConceptNet Numberbatch 19.08* vectors. Originally, it is required to continue to train the model with sentences including these missing multi-word grams. However, 7.5% of the annotated concepts can not find entries, only one of which is the basic level. The missing concepts can be eliminated from the dataset for convenience and the rest 776 concepts (174 of them are the basic level) will be the training data.

Because the benchmark classifier used is Random Forest with SMOTE algorithm, it is not a good idea to feed the 300-dimension vectors into ~~it~~ directly. The reason is that the vectors contain semantics implicitly, unlike the structural features and the frequency features explicitly show the attributes. There would be two ways to represent such semantics. One is to adopt the vectors directly which means they could be learnt by training a Support Vector Machine as a classifier.

However, ~~SVM is turned out from our experiments that performance of~~ classifying the basic level is not as good as Random Forest. Using SVM as the classifier, results from vector-based features did not increase as expected, which ~~is the same situation as~~ Chen's (1). The embedding features trained by the SVM even decrease the performance compared to trained by Random Forest. One possible reason is that it tends to classify the concepts with the similarity of lemmas. In other words, the concepts at the subordinate categories would be more likely to share a higher similarity and form support vectors in the SVM. This ~~could be~~ harm to the binary classification task of the basic level.

Alternatively, with feature engineering, semantic features could be represented by distance. An aggregation method is put forward to extract the semantic features. A lemma distance is firstly defined by the cosine similarity of vectors of two lemmas. The concept distance is then defined by the lemma distance of lemmas in the concept and its hypernyms. And they are aggregated by mean, minimum, maximum, and standard deviation. For example, here is a hierarchy of three concepts in Figure 4.2 to calculate semantic features from their word embeddings. If semantic features of Concept *adjustable wrench* is required, cosine similarities as lemma distances between Lemma *wrench* and *adjustable wrench*, *wrench* and *adjustable spanner* as well as lemma distances between *spanner* and the other two lemmas are calculated. Then the mean, minimum, maximum, and standard deviation of the four lemma distances can become concept distances to represent the semantics.

The semantic features from word embeddings by Word2vec are the four concept distance aggregations of a concept. To find out whether distance-based semantic features improve the performance, we train the benchmark classifier Random Forest with SMOTE algorithm ~~to predict the basic level~~.

4.4.2 Generate From BART

Besides extracting semantic features from word embeddings, we suppose that concepts could be characterized ~~by some words called~~ cues. The cues of a concept would give properties, categories, attributes, and some characteristics to the concept. Inspired by

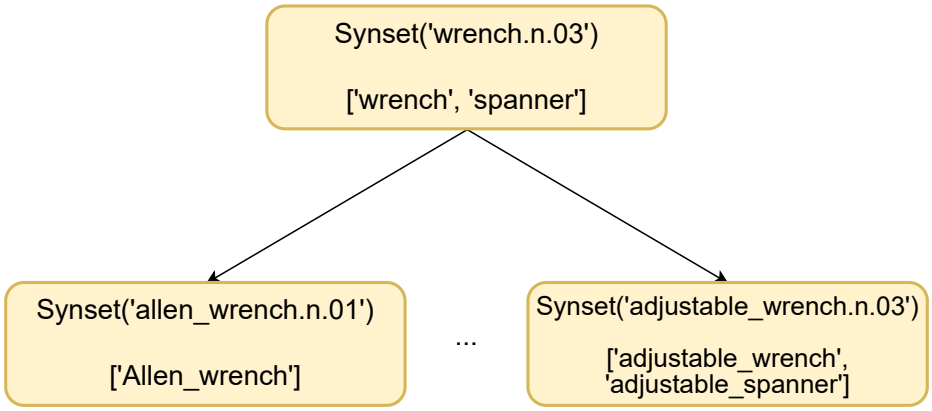

Figure 4.2: Example of calculating semantic features

Machine Translation, these textual semantic features can be generated by a sequence to sequence Machine Learning task. One famous per-trained model BART provides a good tokenization tool as well as a base model to fine-tune for the semantic feature generation.

The training data for fine-tuning is from English semantic feature database (4). The original features of concepts are in different grammar forms. We first do lemmatization to convert them back to dictionary forms. And we only take out concepts and their corresponding lemmatized features to build up the fine-tuning dataset. To make the fine-tuning easier, we define a class of dataset inherited from Torch Dataset [1] to wrap the processed data into a set of dictionary. Each dictionary would be a mapping from a word to its semantic features. The dataset class is also implemented with getLength and getItem functions.

After the processing and wrapping of the fine-tuning dataset. The generation contains three phases. They are tokenization, fine-tuning, and generating illustrated in Figure 4.3.

For the tokenization, as discussed in Section 2.1.4, we use Initialized Encoder to tokenize concepts. BART is here used as a tokenizer to obtain a token identification of each concept and its semantic features. The token identification is in form of a real-number tensor. It encodes text-based information into tensor-based numerical information which contains semantics pre-learned in BART.

For the phase of fine-tuning, following the architecture of BART, we build a sequence-to-sequence trainer to learn the mapping in the English semantic feature database. The hyper-parameters ~~can be seen in~~ Table 4.1. The metric is SacreBLEU which provides BLEU scores used to evaluate Machine Translation models (37). We aim to save the best

[1] `https://pytorch.org/docs/stable/data.html#torch.utils.data.Dataset`

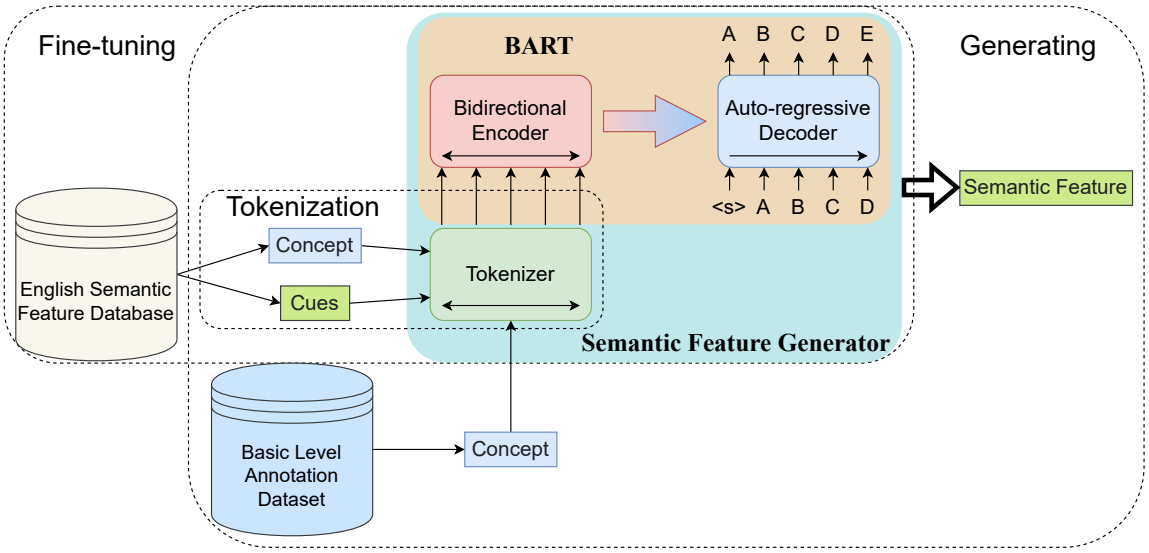

Figure 4.3: Semantic Feature Generation Pipeline

two fine-tuned models during training. The two ~~will be~~ used to translate lemmas into their semantic features.

For the phase of generating, using one of the best fine-tuned model, we can generate semantic features of the concepts in the annotation dataset. The concepts in the dataset can be fully fitted in this pipeline for the semantic feature generation.

Although we are able to generate semantic features, they are text-based which are not easy to learn directly. It is more reasonable to transform the textual feature into a numerical one which ~~is~~ could utilize the semantics. I come up with an indicator which reflects ~~to~~ the original basic level theory, namely cue validity. According to Rosch statement, cue validity

Hyper-parameter	**Value**
evaluation strategy	*after epoch*
learning rate	$2e-5$
train batch size	8
evaluate batch size	8
weight decay	0.01
checkpoint number	2
train epochs number	3
predict with generate	*true*

Table 4.1: Hyper-parameter Setting for Fine-tuning BART

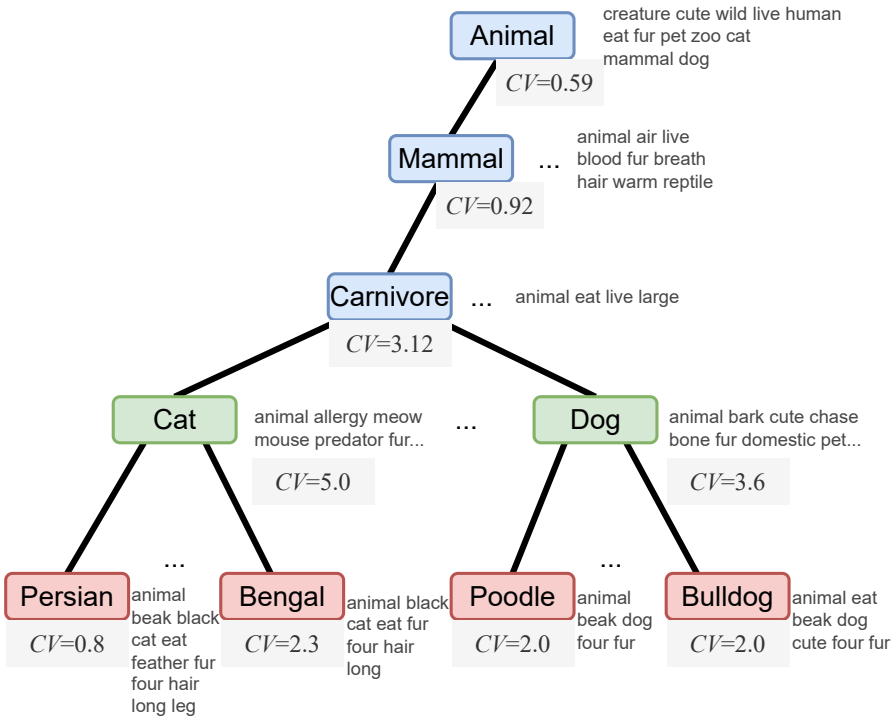

Figure 4.4: Concept hierarchy with cue validities example(4)

can be a probabilistic indicator which the validity of a given cue as a predictor of a given category, ~~in this project is~~ the basic level category. To make it easy to understand, cue validity is from conditional probability to indicate how likely it would be at the basic level.

Discussed in Section 2.1.1, there are ~~various formulas~~ to calculate cue validities. ~~To have a measurement for the cue validity in the project, based on the formal probabilistic conception, here is a formula for compute cue validity given the cue and knowing whether a concept is in basic level~~:

$$val(cue) = P(BL|cue) = \frac{P(BL \wedge cue)}{P(cue)} \tag{4.1}$$

Since a concept does not have the only cue in most cases, the cue validity of one concept, $CV(concept)$, is defined by a sum of the cue validities of a group of cues, which are the attributes of lemmas of the concept, with Equation 4.1:

$$CV(concept) = \sum_{l \in lemmas(concept)} \sum_{cue \in attribute(l)} \frac{P(BL \wedge cue)}{P(cue)} \tag{4.2}$$

The cue validity of a concept is no longer a probability but an accumulation of probabilities instead. The same hierarchy can be taken as an example of defining basic level by

cue validity of concepts, in Figure 4.4. Here are the lemma attributes and a cue validity of each concept in the hierarchy. The cues are mostly from a database by (4) or generated by Section 4.4.2 if concepts are not in the database. It reveals that the concept of cat and the concept of dog show the greatest two cue validities which indicate they are in the basic level. The result keeps the same as that in the approach of cognitive economy.

The number of the cues and the cue validity can be the semantic features of a concept according to the method. These two features are input to train the benchmark Random Forest classifier for predicting the basic level.

5

Experiment Setting

This section states the setting of three models and design of experiments for the research questions.

5.1 Dataset and Model Setup

The basic level annotations and the synthetics features of the concepts constitute the final dataset to be used to train the Random Forest classifier. Three types of models will be performed to test improvement of predicting basic level from the synthetic features. In this section, metrics measuring performance of models, splitting of the dataset, and the three models will be discussed.

Cohen's kappa score(35) and balanced accuracy score(38) are used as metrics for evaluating performance of the models with imbalanced data. Cohen's kappa measures the inter-rater reliability for basic level or not. Specifically, it indicates how well the model predict the basic level correctly compared to predicting randomly by chance. Balanced accuracy is useful to evaluate how good a binary classifier is when trained with imbalanced data. It considers sensitivity, which is the true positive rate, and specificity, which is the true negative rate.

For each experiment, 10-fold cross validation is used to train and evaluate the model. Due to the imbalance of the annotation data, Stratified K-fold[1], a variation of K-fold cross validation which samples each set to contain the same percentage of the basic level as the whole dataset, is implemented to split the dataset into training data and validation data with 10 groups. Under this setting, every experiment will return 10 groups of Cohen's kappa

[1] `https://scikit-learn.org/stable/modules/cross_validation.html#stratified-k-fold`

	Feature type
EX_F_1	Structural features
EX_F_2	Structural features + Frequency features from Google Ngram
EX_F_3	Structural features + Semantic features by Word embeddings
EX_F_4	Structural features + Semantic features by BART

Table 5.1: Experiment Settings for Feature Effectiveness

	Training Data	**Testing Data**
GlobalModel	All 5 domains	All 5 domains
LocalModel	*hand tool*	*hand tool*
	edible fruit	*edible fruit*
	musical instrument	*musical instrument*
	furniture	*furniture*
	garment	*garment*
TransferModel	*edible fruit, musical instrument, furniture, garment*	*hand tool*
	hand tool, musical instrument, furniture, garment	*edible fruit*
	hand tool, edible fruit, furniture, garment	*musical instrument*
	hand tool, edible fruit, musical instrument, garment	*furniture*
	hand tool, edible fruit, musical instrument, furniture	*garment*

Table 5.2: Model Settings

scores and balanced accuracy scores. The two averaged values of the scores respectively are used to evaluate performance of the model.

The experiments include the effectiveness of each type of features based on the structural features, Table 5.1. To verify the method and the synthetic features within and across domains, three models are set up for evaluating and comparing the performance. They are named GlobalModel, LocalModel, and TransferModel, described in Table 5.2.

5.1.1 GlobalModel

GlobalModel is trained and tested with the data from all the five domains in Table 3.1. This model can fully use the annotated concept we have. In other words, it can have as much

data as possible to participate in the training. By GlobalModel, the overall performance of predicting the basic level in a hierarchy will be revealed. The results would indicate whether it is effective to add the synthetic features in training and how much it improves or hurts the accuracy for all the domains.

5.1.2 LocalModel

LocalModel is trained and tested the classifier with the concepts in the same domain. Therefore, there will be five LocalModels trained during one experiment. The result of each LocalModel will indicate whether it is effective to train with the synthetic features within a certain domain. The results of the five domains can then be averaged only to show the influence introduced by the different kinds of features on the five domains. Feature importance in every Random Forest classifier is also returned for comparing contributions of the synthetic features among different domains.

5.1.3 TransferModel

TransferModel is trained with concepts within four of the five domains and tested on the rest one. Similar to LocalModel, there will be five TransferModel trained during an experiment. However, it does not need to set up the 10-fold cross validation because training data and validation data have been splitted by the definition. Result of each experiment is the averaged metrics from TransferModels of the five domains. The aim of TransferModel is to verify generalization of the method. The accuracy on unseen domains means whether the trained model is appropriate to predict the basic level on other domains of knowledge. It can help to detect the basic level in a large-scaled hierarchy, all concepts in WordNet.

5.2 Wilcoxon Rank-Sum Test

To answer the first research question about the relation between prediction performance and the size of corpora, it requires finding whether there is a dependency between the corpus size and the metrics, Cohen's kappa or balanced accuracy. Wilcoxon rank-sum test, also known as Mann-Whitney U test(39), is performed to test the null hypothesis that the prediction performance Cohen's kappa values and balanced accuracy by different sizes of the same corpus are from the same distribution.

The experiment focus on the source of frequency features. Before conducting Wilcoxon rank-sum test, the corpora are sampled into different sizes in Table 5.3 and used to calculate

Corpus	1M	2.4M	5.7M	100M
BNC	√	√	√	√
CHILDES	√	√	√	
CABNC	√	√		
KBNC	√			

Table 5.3: Corpora Sampling in Different Sizes

frequencies. Each corpus in the scheme is sampled 50 times. For example, BNC is sampled into the word counts of 1 million, marked *BNC 1M*, 50 times. Therefore, there are 500 sampled corpora in total including 50 *BNC 1M*, 50 *CHIDES 1M*, 50 *CABNC 1M*, 50 *KBNC 1M*, 50 *BNC 2.4M*, etc. The sampled corpora will be the sources of the frequency features which are used to train the Random Forest classifier.

After sampling, the classifier is trained and tested by the structural features and the frequency feature. With the model setting in Section 5.1, frequency features from different sampled corpora can be used to train and test with the three models. Each corpus in a specific size leads to 50 results with each model. Totally, there will be 1500 groups of Cohen's kappa and balanced accuracy results from the corresponding models and frequency sources.

Wilcoxon rank-sum test is carried out to test two aspects of hypotheses about the size and the type of corpora. The experiment settings are described in Table 5.4 and Table 5.5. The initial setting EX_W_0 is to have the Cohen's kappa and the balanced accuracy of each model from the samplings in Table 5.3. The first setting EX_W_1 is to compare the results from the same corpus but different sizes. The second setting EX_W_2 is to compare the results from the same discourse type of a corpus but different sizes. The third setting EX_W_3 is to compare the results from the same target audience of a corpus but different sizes. The last setting EX_W_4 is to compare the results from the same size but different discourse types and target audiences of corpora.

	Corpus	Size
EX_W_0	KBNC	1M
	CABNC	1M, 2.4M
	CHILDES	1M, 2.4M, 5.7M
	BNC	1M, 2.4M, 5.7M, 100M
EX_W_1	CABNC	1M - 2.4M
	CHILDES	1M - 2.4M
		1M - 5.7M
		2.4M - 5.7M
	BNC	1M - 2.4M
		1M - 5.7M
		1M - 100M
		2.4M - 5.7M
		2.4M - 100M
		5.7M - 100M
EX_W_2	Written	1M - 2.4M
		1M - 5.7M
		1M - 100M
		2.4M - 5.7M
		2.4M - 100M
		5.7M - 100M
	Spoken	1M - 2.4M
		1M - 5.7M
		2.4M - 5.7M
EX_W_3	General	1M - 2.4M
		1M - 5.7M
		1M - 100M
		2.4M - 5.7M
		2.4M - 100M
		5.7M - 100M
	Children	1M - 2.4M
		1M - 5.7M
		2.4M - 5.7M

Table 5.4: Experiment Settings for Wilcoxon Rank-Sum Test

	Size	Discourse Type / Target Audience
EX_W_4	1M	Written - Spoken
		General - Children
	2.4M	Written - Spoken
		General - Children
	5.7M	Written - Spoken
		General - Children

Table 5.5: Experiment Settings for Wilcoxon Rank-Sum Test

6

Results & Evaluation

Discuss the design of your experiments, the results you obtained, and how they help in evaluating the claims you made in the introduction. You may also use the evaluation results in this section to justify your design choices or assess the contributions of different aspects of your design towards the overall goals.

7

Discussion

Here you put your results in context (possibly grouped by research question). Usually, this section focuses on analyzing the implications of the proposed work for current and future research and for practitioners.

8

Conclusion

Briefly summarize your contributions, and share a glimpse of the implications of this work for future research.

References

[1] Yiwen Chen and Simone Teufel. **Synthetic Textual Features for the Large-Scale Detection of Basic-level Categories in English and Mandarin**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8294–8305, 2021. iii, 5, 12, 27

[2] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. **Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. *arXiv preprint arXiv:1910.13461*, 2019. iii, 9, 10

[3] George A Miller. **WordNet: a lexical database for English**. *Communications of the ACM*, **38**(11):39–41, 1995. iii, 11, 13, 14

[4] Erin M Buchanan, Kathrene D Valentine, and Nicholas P Maxwell. **English semantic feature production norms: An extended database of 4436 concepts**. *Behavior Research Methods*, **51**(4):1849–1863, 2019. iii, 18, 26, 28, 30, 31

[5] Roger Brown. **How shall a thing be called?** *Psychological review*, **65**(1):14, 1958. 3

[6] Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. **Basic objects in natural categories**. *Cognitive psychology*, **8**(3):382–439, 1976. 3, 4, 22, 23, 24

[7] Lala Hajibayova. **Basic-level categories: A review**. *Journal of Information Science*, **39**(5):676–687, 2013. 3

[8] George Lakoff. *Women, fire, and dangerous things: What categories reveal about the mind*. University of Chicago press, 2008. 3

[9] Rebecca Green. **Vocabulary alignment via basic level concepts**. *Final Report*, 2003. 3

[10] David J. Finton. *Cognitive-Economy Assumptions for Learning*, pages 626–628. Springer US, Boston, MA, 2012. 4

[11] Lee Roy Beach. **Cue probabilism and inference behavior.** *Psychological Monographs: General and Applied*, **78**(5-6):1, 1964. 6

[12] Lee Roy Beach. **Recognition, assimilation, and identification of objects.** *Psychological Monographs: General and Applied*, **78**(5-6):21, 1964. 6

[13] Stephen K Reed. **Pattern recognition and categorization**. *Cognitive psychology*, **3**(3):382–407, 1972. 6

[14] Leo Breiman. **Random forests**. *Machine learning*, **45**(1):5–32, 2001. 7

[15] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. **SMOTE: synthetic minority over-sampling technique**. *Journal of artificial intelligence research*, **16**:321–357, 2002. 7, 8

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. **Distributed representations of words and phrases and their compositionality**. *Advances in neural information processing systems*, **26**, 2013. 8

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. **Efficient estimation of word representations in vector space**. *arXiv preprint arXiv:1301.3781*, 2013. 8

[18] Robyn Speer, Joshua Chin, and Catherine Havasi. **ConceptNet 5.5: An Open Multilingual Graph of General Knowledge**. pages 4444–4451, 2017. 8

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. **Attention is all you need**. *Advances in neural information processing systems*, **30**, 2017. 9

[20] Chad Mills, Francis Bond, and Gina-Anne Levow. **Automatic identification of basic-level categories**. In *Proceedings of the 9th Global Wordnet Conference*, pages 298–305, 2018. 10

[21] Laura Hollink, Aysenur Bilgin, and Jacco van Ossenbruggen. **Predicting the basic level in a hierarchy of concepts**. In *Research Conference on Metadata and Semantics Research*, pages 22–34. Springer, 2020. 11, 15, 21, 22, 24

[22] Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. **Syntactic annotations for the google books ngram corpus**. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174, 2012. 11, 17

[23] Niamh Henry. *Learning the basic level from text: Studying different corpus characteristics in predicting the basic level*. PhD thesis, 2021. 11, 21, 22, 23, 24

[24] Marco Baroni and Alessandro Lenci. **Distributional memory: A general framework for corpus-based semantics**. *Computational Linguistics*, **36**(4):673–721, 2010. 12

[25] Wen-hao Chen, Yi Cai, Ho-fung Leung, and Qing Li. **Context-aware basic level concepts detection in folksonomies**. In *International Conference on Web-Age Information Management*, pages 632–643. Springer, 2010. 12

[26] Ciro Cattuto, Dominik Benz, Andreas Hotho, and Gerd Stumme. **Semantic grounding of tag relatedness in social bookmarking systems**. In *International semantic web conference*, pages 615–631. Springer, 2008. 12

[27] Mark Gluck. **Information, uncertainty and the utility of categories**. In *Proc. of the Seventh Annual Conf. on Cognitive Science Society, 1985*, 1985. 12

[28] Saul Albert, LE De Ruiter, and JP De Ruiter. **CABNC: The Jeffersonian transcription of the spoken British national corpus**, 2015. 16

[29] Brian MacWhinney. *The CHILDES project: Tools for analyzing talk, Volume II: The database*. Psychology Press, 2014. 16

[30] BNC Consortium et al. **British national corpus, XML edition**. *Oxford Text Archive. http://hdl. handle. net/20.500*, **12024**:2554, 2007. 16

[31] Brian MacWhinney. **Tools for analyzing talk part 1: The chat transcription format**. *Carnegie.[Google Scholar]*, 2017. 16

[32] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Google Books Team, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, et al. **Quantitative analysis of culture using millions of digitized books**. *science*, **331**(6014):176–182, 2011. 17

[33] Martin F Porter. **Snowball: A language for stemming algorithms**, 2001. 18

[34] Gregory L Murphy and Edward E Smith. **Basic-level superiority in picture categorization**. *Journal of verbal learning and verbal behavior*, **21**(1):1–20, 1982. 22

[35] Jacob Cohen. **A coefficient of agreement for nominal scales**. *Educational and psychological measurement*, **20**(1):37–46, 1960. 25, 33

[36] Allan M Collins and Elizabeth F Loftus. **A spreading-activation theory of semantic processing.** *Psychological review*, **82**(6):407, 1975. 26

[37] Matt Post. **A Call for Clarity in Reporting BLEU Scores**. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. 28

[38] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. **The balanced accuracy and its posterior distribution**. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE, 2010. 33

[39] Henry B Mann and Donald R Whitney. **On a test of whether one of two random variables is stochastically larger than the other**. *The annals of mathematical statistics*, pages 50–60, 1947. 35

Appendix