Formal Grammar Generating Procedure for Domain Specific Vocabulary Applications and Optimising Speech Recognition and Voice Control.

Shashwat Goel, Delhi Public School RK Puram, New Delhi
Wrik Karmakar, South City International School, Kolkata

Abstract

We develop a novel approach for speech recognition and control with increased robustness, tolerance to failure and applicability to critical domains like aviation, security, space, and healthcare where training regimes exist. Furthermore, recognition systems independent of English grammar are immensely useful for non-native speakers, who then just need to memorize a smaller vocabulary and its meaning. We propose an automated method of generating concise sets of user-friendly, unambiguous task-oriented commands from text corpora, to improve discernibility.

We pre-process varying datasets to produce uniform input. Customized Part-of-Speech tags are used to suit whatever industry required. Lexicons for each tag are processed separately hereafter. Word embeddings provide a numerical visualization of this semantic interpretation of words. Thus, each lexicon separated for different commands is plotted in a 300-dimensional vector space by training embeddings using a GLoVe system enriched specifically by troponym relations, a feature of verb pairs that most accurately describes verb relations. K-means clustering is performed to group words with similar meaning. The 'K' to run K-means clustering is chosen by the elbow method, that is, a F-Test (Number of clusters vs Percentage variance) graph.

Each cluster is represented by a central word, and the clustering condenses the vocabulary. To achieve this we consider 5 parameters - vocabulary size, phonetic distance, loss of meaning, familiarity, and command complexity. Among these the most innovative was avoiding phonetic collisions between selected words. Phonetic similarity was evaluated by ABX testing, traditionally used to evaluate compression quality.

For this, a congruence to the NP-Hard Maximum-Independent Set problem was proven by taking words as nodes and augmenting edges between phonetically or semantically similar pairs. To convert from a node-weighted graph to an unweighted one, we proved that if a particular node is selected, all its copies can be selected for a better solution. Consequently we also proved that the weight is accounted for in this process. Observing the sparsity of the graph, a SAT Solver is used. Rapid Automatic Keyword Extraction (RAKE) was used on WordNet definitions to represent the remaining words in the cluster, ensuring minimal loss of semantic meaning.

The vocabulary and syntax are finally to be parsed into a grammar-assisted speech recognition model. We propose an innovative method for evaluation. The dataset and its formalized version are converted from text-to-speech with noise added from the AURORA database. This speech is then converted back to text, and accuracy compared.

We have currently applied the procedure on the Cornell Tennis Commentaries dataset. A 53% reduction in the Noun lexicon, and 37% reduction in the Verb lexicon was achieved. Given that the dataset is not a command/control dataset, the results will only improve once our request to access such private datasets is approved. We hope to create a custom speech recognition using the CMU Sphinx software to fully test our work in the future

Giving a computational basis to what has previously been attempted manually to create callsigns and radiotelephony procedure, we empower the revolution of intelligent systems that our future shall embrace.

Introduction and Objectives

Around four hundred kilometers above the surface of the earth, in the floating research center we know as ISS, a language called Runglish was born, purely out of necessity. The term attained popularity in 2000, when in an interview, a cosmonaut who was the inhabitant of the ISS, Sergei Krikalyov said: "We say jokingly that we communicate in 'Runglish,'[1] a mixture of Russian and English languages, so that when we are short of words in one language we can use the other, because all the crew members speak both languages well."

Recent developments in the private space industry have made leaps and bounds towards making space colonization and exploration a reality. This coupled with amendments in UN Space Policy is encouraging a newfound diversity in the organizations and countries engaging in this mission. While this brings immense advantages in terms of the resources and technology available, there is an added layer of complexity that must not go unnoticed - the days when balancing Russian and English seemed to be enough for comprehension are drawing to an end.

Human interactions shall perhaps nevertheless be satisfied with varying accents of English (and a little patience), but the only way space exploration will be possible on a commercial scale is through proactive development of intelligent robotic systems. The current design of spacesuits imposes constraints on human mobility, and scarce resources limit extravehicular missions. A solution that we came across was of course speech recognition. Speech control would go a long way in optimizing both computers and robots, and smart visors which after Google Glass are no more considered futuristic could actually find use in space. So, why not speech recognition?

This was the question that hit us while we were working on a project to design settlements in Space. Of course, well documented answers exist. The reason is that it cannot be relied upon. The accuracy and speed of current speech recognition systems make them unsuitable for time critical operations like those in space. Errors that are humorous here are fatal there. While speech recognition optimization continues at a slow but steady rate, it is doubtful that a level of trust will be built soon enough. And even if we got past the inaccuracy barrier, what language would speakers unfamiliar with Russian and English both use to address those intelligent robotic systems, and how quick would that be?

The scenario prompted us to come up with a different approach to enhancing speech recognition. We start with the hypothesis that: *A formalized grammar limited to domain specific vocabulary generated through a uniform semi-automated procedure significantly improves accuracy and speed of speech recognition.* It is founded upon the now widely accepted fact that with a smaller vocabulary size and formal grammar, the recognition accuracy is significantly superior to that obtained with a similar system trained on a free form grammar. This has been illustrated on multiple instances, notable of which is [Kaljurand and Alumae, 2012][2] who showed that accuracy improved from 60% to 90% by using a formal grammar for recognizing google maps addresses. This is primarily because a limited vocabulary to choose from intuitively means probabilistic models will perform better. Speech recognition on a restricted vocabulary and formalized grammar of commands minimizes chances of recognition based errors thus allowing for a much more reliable system irrespective of the speed-accuracy tradeoff. Our approach remains relevant even if improvement in learning models over time make speech recognition sufficiently efficient for

[1] "Russian language-Runglish - Languages on the Web." Accessed December 30, 2018. https://www.lonweb.org/links/russian/lang/020.htm.

[2] "LNAI 7427 - Controlled Natural Language in Speech ... - Springer Link." https://link.springer.com/content/pdf/10.1007%2F978-3-642-32612-7_6.pdf.

use in such industries, as recognition on a limited vocabulary size will always outperform the same model's performance on a larger vocabulary.

We propose a formalized grammar selection and implementation system, that given a dataset of the human speech commands spoken during the concerned application was being executed helps generate a formal set of rules and word classifications. These are combined into a grammar which is required to be followed by users. The generated grammar is automatically fed into the required recognition systems. While requiring users to learn a command set may seem counter-productive and a primitive approach at first, it is widely used by security agencies across the world. The lack of a mathematical basis to the current military speech models clearly shows the need for our system. There are obvious examples of where current military speech systems fail. Moreover, since security related domains entail the need for a different callsign and command list for different agencies, our proposal provides maximum benefits to this field.

However it would be a mischaracterisation to say that our system is limited to space applications. We believe it has far reaching applications in fields like healthcare operations, aviation, and crucially, robotics - all fields where the issues we found in space appear in one form or the other.

Speech Recognition has applications in prescriptions (inaccuracies in pharmeticians understanding handwriting and caretakers identifying complex drug names have been a proven cause of incorrect dosage as outlined by Omoregbe in *A Voice-based Mobile Prescription Application for Healthcare Services (VBMOPA)*) and discharge summaries [SB Johnson wrote about the need for speech recognition interfaces for discharge summaries in '*A Semantic Lexicon for Medical Language Processing,' 1999*[3]] apart from robotic assistance in surgical operations. The accuracy of the recognition can significantly improve with the use of a formalized grammar, which is easy to formulate using our algorithm in this particular context due to the existence of vast corpora of textual records.

Limited vocabulary recognition has been proven to be beneficial especially in aviation where cockpit speech is often interfered by noise [C Englund, 2004][4]. As we move towards making aircrafts more capable of independent flying with less human oversight, this is a great push forward in that direction. The applications of our model in robotics are perhaps most easily explained. Robots can be easily addressed by a system of expressions having a formal grammar. This requires no additional effort from the user, especially in commercial systems, as the guidance manuals help them in training stages and later they are habituated to the newer method of communication. Moreover, while this paper focuses its approach on the English language, only minimal assumptions and observations have been made keeping just English in mind, and even those can easily be adapted for different languages. The only necessity is versions of the used algorithms and tools to be available in the chosen language. Notably, in all these industries, the users undergo long training programs which should be sufficient for mastering the required commands, especially as the syntax can easily be close to the chosen language. Thus the question of going backwards towards primitive technology does not hold.

We thus outline an adaptable general procedure and do not limit our scope to any specific industry.

---

[3] "A semantic lexicon for medical language processing. - NCBI - NIH." https://www.ncbi.nlm.nih.gov/pubmed/10332654.

[4] "Speech recognition in the JAS 39 Gripen aircraft - Division of Speech ...." 11 Mar. 2004, http://www.speech.kth.se/prod/publications/files/1664.pdf.

Application specific datasets are instead used for demonstrative purposes.

A common counterargument that we were posed by our immediate reviewers was that, in the interest of accuracy, one might as well incorporate buttons which makes the process mechanical rather than intelligent and hence obtains for the best possible accuracy on the machine end. The several scenarios in which case buttons prove not less useful but utterly useless as compared to speech recognition systems - hands being otherwise engaged as these critical situations often if not always entail multitasking, furthermore coupled with the fact that it is not feasible to carry controllers for multiple devices. Moreover, we prove the scope and viability of our concept by assuming that speech recognition will be predominant, and our system could significantly improve it for a nominal cost in terms of training requirements, which too is an existing process in our target industries.

While there may nevertheless remain operational challenges in installing entire speech recognitions to replace existing interfaces, it is definitely a significant occupation and the effect it has shall only be amplified by the future that is to come. In a world set to be driven by the Internet of Things and automation, our research has the potential to solve the most fundamental problem with the implementation of these mechanisms. Speech Recognition is simply not at the level where it can disambiguate common everyday commands wholly, but we strive to build that bridge through our program. It is admittedly difficult for formal grammar to be imposed for interaction with personal assistants, but the system has immense capabilities. The following are objectives including both, those already achieved and those which are beyond the scope of this paper but are objectives to realize the full potential of the proposed technology:

Objectives:

1. Outline a procedure to generate a fully functional formalized grammar using limited vocabulary for commands that takes a dataset of speech recognition systems communications.
2. The proposed grammar should follow the properties of a good controlled natural language as outlined by Schwitter : "*(a) it should have a well defined syntax and a precise semantics that is defined by an unambiguous mapping into a logic based representation; (b) it should look as natural as possible and be based on a subset of a certain natural language; (c) it should be easy for humans to write and understand and easy for a machine to process; and (d) it should have the necessary expressivity that is required to describe a problem in the respective application domain.*"
3. Extend capabilities and versatility to utilizing instructive/imperative human conversations and extrapolate data for speech recognition systems. This broadens the potential and use-cases.
4. While initially the proposed system will require some human interventions to pick out edge cases and phonetic collisions (when two words prove difficult to discern for the program due to phonetic similarity) in commands, and in selecting entities, improvement in automated technologies for these tasks can be used to replace the currently proposed mechanisms.
5. Since a generalized algorithm is proposed, it can be easily adapted for multiple languages. This requires that the various subproblems like detection of semantic similarity are facilitated by existing data and libraries for the language.
6. The ultimate goal is an open-sourced limited vocabulary platform with trained mechanisms, datasets and vocabularies for all kinds of fixed-purpose robotic speech recognition. With improvement in natural language understanding systems, all robotic tasks will transition towards speech-augmented control and in this scenario it is even more important that recognition inaccuracies don't hamper this revolution.

Innovation

The novelty of our program is underscored best by the approach it takes to optimise recognition - it does not simply quicken or better speech processing but takes an altogether different road by stripping down audio to a sequence of well defined predicates with their specific arguments which is then processed. We have developed a method to map the expressions we use to trigger actions in machines into a comprehensive and precise set of commands that are obtained along the following parameters -making sure the vocabulary is small, that the words are distinct and common to us, and also that the words are good approximations of the extremes of their meaning . We build an alternative for specific industrial or technical usage that sustains a better processing rate by limiting vocabulary, in comparison to large vocabulary corpora. Through this project, we seek a means to reach crucial functions of a machine that are time intensive but fail to be triggered adequately or even accurately enough for solving the problem at hand. We optimise a industry specific speech recognition system to readily understand the action required from a common expression and in the process, optimise speech recognition as a whole, making the bridge between human expression and machine interpretation shorter than ever before. Herein lies the innovation of our idea.

The idea has been entirely ours and we have come across very little work that resembles ours, establishing that our idea, for better or worse, is very much out of the box. Having said so, the implementation of our idea would not be possible without the efficient gears and tools that we borrowed from academic discourse on natural language processing. Although the tools we use are mostly well established instruments in their own right, they have never been married for such a purpose priorly.

Our program in its initial step involves the reorganisation of data for ideal configurations. We use the NLTK POS tagger to identify the various parts of speech and isolate the ones we do not require for our procedure. We go on to process the verbs differently and the rest of the POS in a more standard manner. ###A few of the conflicts we prepared our algorithm to avoid which had little work done on them simultaneously were conflicts of possession, conflicts of negation and antonyms, conflicts of tense, and mapping multiple word verb phrases to single word equivalents. We solved these by making arrangements individually for all of those conflicts and then integrating them together for the purpose of the program. We first find the verb roots for all words by stemming to remove tense markers. We used a standard prefix against a root word to denote the antonyms that would preserve context and negation property both. This was a marked difference from previous work which ether isolates from context to preserve negation, or conceals negation to preserve context.

Once we obtain the appropriate taggings, the next stop down the road is finding a way to group the probable words that convey similar semantic value and can be used interchangeably without drift in meaning. Presently we utilize word embedding systems like GloVe, implementations of which are available freely. However this has caused problems in critical language understanding tasks such as intent detection and slot filling in spoken language understanding as they deal with phrases and sentences rather than individual words. Given that the semantic clustering task is one of the key steps for the efficacy of our proposed procedure, existing cutting edge research can be applied to improve our model. We perform intent detection and place words on a vector space after enriching them significantly to appropriate their usefulness (Kim, Tur, et. al, 2016). It enriched vectors on the basis of (i)similarity with antonyms, (ii) similarity with synonyms, and (iii) similarity to the initial neighboring words(for regularising purposes). However as (Navaretta, 1999) notes, verbal linguistic scales exist, but they are not so frequent as adjectival

scales, and only few verbs have ”real” opposites. This gave us the idea to include the troponymy relation. According to Miller et al. (1993) verbs are troponyms if they are connected to a super-ordinate along more semantic dimensions. On the class of verb concepts two semantic relations are defined which are not logically independent: It is stated in [Miller et al., 1993]) that "Troponymy is a particular kind of entailment, in that every troponym V 1 of a more general verb V2 also entails V2" (p. 47). In simpler terms, an example would be while *'nibble'* and '*eat*' aren't exact synonyms, *nibble* entails *eat*. Thus,for the purposes of our program, we consider verbs to be ”similar” if they belong to a linguistic scale, are opposites, synonyms, or troponyms. We integrate this into the function for vector enrichment as well, thus adding the (iv) factor - similarity with troponyms. This is a major new development that paves the road for our succeeding process. Of course the troponymy relation would have been included while measuring the cosine angles of related connections n the original function as well, however at a much lesser weight than it should have among verbs.

As mentioned before we were attempting to control the vocabulary size while maximising familiarity and distinct nature of the verbs. In order to do so, we took the already segregated clusters and picked exactly one word from each based on minimal distance from the centre of the cluster, terming them leading nodes. This solved our problems of familiarity, meaning, as well as vocabulary size. The method of ensuring distinction in terms of pronunciation is however where we made the most radical approach towards solving a problem. We developed a graph analog for the problem statement, using a threshold of ABX Testing to draw edges between pairs of words that sound same. After accounting for weights and solving accordingly we arrive at a reduced problem statement of finding the maximal independent set of vertices of the graph. This forms our core solution.

The application of ABX testing to phonetic disambiguation is a very elegant solution to an otherwise complicated problem (some of the other solutions are briefly described in the algorithm section). ABX testing is traditionally used to disambiguate between compressed and original audio samples. It involves playing A, then playing B, and then playing multiple random samples X asking a human to classify them as A or B. If the human can do this accurately, then the compression algorithm is not good enough, where as if the difference is subtle the compression algorithm has achieved its goal.

Inspired by this, we pick a pair of 2 words say C and D. We then use speech generation softwares to utter C and D as part of some sentences in varying tones and accents. A speech recognition system works by using Hidden Markov Models, such that each possible outcome is given a probability. We compare the probabilities assigned by the speech recognition system to the utterance being C and D. We define an accurate guess as an assignment of higher probability to the word that was actually generated using the software. If the number of inaccurate guesses is sufficiently low (an appropriate threshold can be set based on the application) then the two words will not be confused by the recognition system at the time of real use. Thus, an edge need not be augmented between them.
This system thus does not try to assign and decipher from numerical context, rather testing the physical properties of the system itself.

For representation of words besides the predicates in their respective clusters we develop a strategy that skirts past the problem of word analogies - an extremely understudied and is an unsolved problem in NLU. Word analogies would be an obvious heuristic as they fit the problem specifications well, however the difficulty of the problem forced us to look elsewhere for solutions. We resorted to Wordnet short definitions of words to solve this as these definitions are less than 4 words long in most cases. These words are mapped to the most semantically similar

components of the vocabulary, and the resulting mappings are then passed onto the next step for syntactic arrangement.

In addition to these specific features, we also developed an interesting testing procedure instead of using a cost function to analyze accuracy based on abstract parameters. We compare the accuracies of the speech to text conversion of the input data set with the corresponding sequences of predicates, predicate-manners, and arguments, while also checking for vital entity loss using a paraphrase detector. The speech to text conversion is also conducted using varying noise samples from different industries (such as helicopter noise samples) so as to reaffirm our findings for speech recognition robustness properties.

This is incidentally also the first work where we see development of formalised grammar for a natural language in speech recognition purposes. While some work has been done towards constructing a formal grammar they have been focused on proving the viability and need of such systems. No work exists to actually generate a formal grammar from traditional datasets. Developing such a system for speech recognition makes the possibility of transformation to speech recognition based communication channels for attractive than ever before especially for sectors which have concerns about ambiguity being fatal to their purpose such as healthcare. As the popularity speech recognition grows as a technology grows by the day, presenting an alternative heuristic for solving several key accuracy concerns of its helps us contribute to its rise.

Algorithm

Our system requires skilled supervision and human intervention at certain steps to ensure accurate results. Being a largely one-time use system, it is meant to facilitate the generation of the formal grammar, not fully do it by itself. This it does by classifying words into syntactic classes and nouns into semantic classes so that they can be broadly distributed into 'predicates' and their 'argument types' with a limited but connected lexicon for each. What this means is that if 'John' and 'Jane' only occur as 'Agent' and 'Patient', the algorithm should be able to connect that these entities can occur as both even if the dataset does not provide sufficient examples. Similarly, if 'Fast' occurs only as manner argument of 'eat' in the dataset, it should still be a potential argument of sleep. This is achieved by grouping all 'Manners' into a single set of arguments and checking for these.Providing a list of semantic entities, performing the various testing and algorithm procedures, and setting the syntax for the final output commands will be the main role of the human operator involved.

Our algorithm can be broken down into the following steps:

1. Pre-processing Input - Tokenizing
2. Classification of Tokens into Predicates and their Arguments, creating sub-dictionaries of each argument type.
3. Clustering of tokens within each sub-dictionary on the basis of semantic meaning
4. Selecting cluster representative words for vocabulary by simultaneously optimizing parameters
5. Representing whole input dataset sentences in terms of chosen vocabulary words by combining words, using arguments and specifying intensity parameters.
6. Establishing syntax for order of commanding verbs and arguments to formalize grammar.
7. Training speech recognition model accordingly and providing final grammar as output to user.

We present detailed analysis of each of these steps.

1. Pre-Processing

It is needless to state that a multi-faceted algorithm like that outlined incorporating a wide variety of algorithms in each step would require a exhaustive dataset for generating the most optimal formal grammar required. The data required consists of instructive sentences or queries which are somewhat relevant to communication between humans and speech recognition systems. The procurement of a sanitized (noise cleaned and recognition errors corrected) text corpus is required before pre processing. The grammar will be produced for each application independently. Our algorithm must be able to custom cater to a particular task taking part in security operations as the functions of each differ. This causes a problem. It may not always be feasible to depend on datasets custom made for our program as truly catching all possible phrases said to it may take a very long time despite the dataset being able to identify argument classes, members of which can be used replacably in their contexts.

Thus, our algorithm must be able to utilize instructions given by users to humans as well as recognition systems. This is because our target industries might be ones that have very few if any established dedicated speech recognition systems in the first place. Capturing intent and key words from human conversations and dialogues is a problem that has not been solved accurately and continues to merit attention. However, it has been shown that using imperative 'command and control' sentences for extracting semantic meaning is extremely helpful. This is because of the lower probability of troublesome elements like conjunctions and negation particles. This has been shown extensively in the recent work in Grounded Language Learning.[Matuszek 2018], [Pillai 2018], [H Yu, 2018] etc.

Careful analysis of our target industry shows that this forms the basis of an important exploit. Industries such as the military heavily depend on instructions issued by superiors. An existing speech recognition system can be deployed to collect voice samples over a certain time period. This data can then be pruned to acquire the required datasets. A similar approach can be used in all areas our algorithm needs to be applied. This minimizes the required human labour while providing a usable dataset.

After the accuracy of the speech to text conversion, or directly existing text dataset is confirmed, to utilize a fairly uniform system for varying applications, the issued instructions need to be brought into a similar form irrespective of the type of dataset.

First, all verbs are stemmed into their base forms by using a stemmer (the choice of stemmer was arbitrary). This is because tense information is not necessary for commands and queries. Instructions to speech recognition are more likely to contain specifics on the time where it is relevant.

The dataset sentences need to be converted into imperative speech. The NLTK POS Tagger is used to transform words into word tokens that also include their part of speech tags. The same word but in a different part of speech context tag is considered as separate for the purpose of the rest of the algorithm. The following is a list of NLTK tagsets Tags.
The tags whose words are used as they are as follows:

CC	coordinating conjunction	
CD	cardinal digit	
DT	determiner	
FW	foreign word	
IN	preposition/subordinating conjunction	
JJ	adjective	'big'
NN	noun, singular	'desk'
NNS	noun plural	'desks'
NNP	proper noun, singular	'Harrison'
NNPS	proper noun, plural	'Americans'
WDT	wh-determiner	which
WP	wh-pronoun	who, what
WP$	possessive wh-pronoun	whose
WRB	wh-abverb	where, when
VB	verb, base form	take
RB	adverb	very, silently,

The tags either whose words are irrelevant or which may not come up in our dataset should have these words deleted. They are as follows:

EX	existential there (like: "there is" ... think of it like "there exists")	
LS	list marker	1)
MD	modal	could, will
PDT	predeterminer	'all the kids'

The tags which are handled in a different manner are as follows:

JJR	adjective, comparative	'bigger'
JJS	adjective, superlative	'biggest'
POS	possessive ending	parent's
PRP	personal pronoun	I, he, she
PRP$	possessive pronoun	my, his, hers
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	particle	give up
VBD	verb, past tense	took
VBG	verb, gerund/present participle	taking
VBN	verb, past participle	taken
VBP	verb, sing. present, non-3d	take
VBZ	verb, 3rd person sing. present	takes

Comparatives and superlatives can be confused quite easily in detection due their phonetic similarity. The 'er' in bigger and better is often pronounced as a ə making it hard to detect. Given this scenario, we will handle this during our pre-processing step itself. Comparatives and Superlatives will be converted into 'base form + comparative marker' and 'base form + superlative marker' respectively. This will be done both for adjectives and adverbs.

Possession in the form of 'X's Y' can be detected wrongly if the 's is not clear. These will be transformed to 'Y of X' in pre-processing. Personal Pronouns and Possessive Pronouns will first be dereferenced by interpreting which entity they indicated. This can be done with a fairly high accuracy as shown by [Durett and Klein, 2013] through their statistical approach based on a large annotated corpus and the novel Sieve based hand coded system proposed by [Lee et al, 2013]. The Stanford DCOREF is an open existing implementation available for the same. This helps identify the 'X' as mentioned in the previous possession system. Now this too can be represented in the Y of X form. The current inaccuracies in interpretation is due to nominal anaphora. Nominal anaphora is connecting words to their antecedents and postcedents when these are names. It can be avoided by our procedure as names of classes like diseases and medicines in the surgical industry have databases that can be uploaded and used directly. This is used again in Step 2 - Classification of Tokens.

Multiple word phrases like 'give up' will be plotted by using word embeddings along-side all other words and then their frequency will be distributed among single words that convey the same meaning (have similar cosine angles). This is because the frequencies play an important role in making an assumption for the Step 4 (Vocabulary Selection step) of the algorithm as outlined above and allocating the whole frequency of the phrase to any of the synonyms or constituent words would create bias towards them in the selection process without having any semantic basis.

Another important aspect is the handling of antonyms. When word embeddings are used in step 3, antonyms will automatically be grouped with synonyms. To avoid this, antonyms are converted into base forms in synonyms. For example, Un_X_ can be converted into OPP_X_ where OPP serves an antonym particle. When the synonym and antonym form 2 distinct base words such as hot and cold, this is not done to prevent difficulty in learning the proposed grammar. This allows the speakers to say both hot and cold, instead of saying hot as 'OPP_COLD_'.This preserves the frequency of the base form while also ensuring that an antonym form does not end up causing problems in clustering and cluster representative word selection (Steps 3 and 4).

This completes the pre-processing on the input dataset to have uniformly formatted data despite varying collection sources so that a standardized algorithm can be used from this point.

2. Classification of Tokens

The classification of tokens is an important problem for the success of our proposal. It is also one that requires input of data other than the main corpus. The complexity of this problem varies from application to application. As an example, using our system for the medical industry would require labeled datasets of tools required, diseases, medicines, procedures etc. These often have terms which have been found to be tough for speech recognition softwares to recognize. Industry specific proper nouns will be classified into different categories based on these inputs provided. This input has to be uploaded by the human operator.

Before classification is studied, it is important to understand that a relatively similar human made format will be followed for the grammar. Our proposed format is based on Predicates and Arguments, much like the HTML language that forms the basis of the Internet. In this system, the verbs of the input dataset sentences will be taken into the

command lexicon. Each predicate will have its own set of arguments. The human operator will select the overall umbrella of headings of the arguments that are required which will then be used to assign each predicate it's own argument-set by the program itself.

It is important to note that this classification of predicates and arguments is to be utilized in the next step. The lexicon of the predicate and 'modifier' argument (those which specify mannerisms in PropBank) will be semantically clustered and from each cluster one word will be chosen to represent the cluster. This rules out the selection of highly interconnected clusters such as those picked using selection preferences on WordNet or frames of FrameNet. Instead, a simpler and more uniform classification like that of PropBank is chosen where each verb has its own sets of arguments. For recognition of proper nouns and their PropBank tag, the industry specific lexicon uploaded by the human operator will be used. The operator can manually control the final list of tags based on the application.

Semantic Role Labelling will then be performed on the input dataset, creating lexicons of the predicates and the 'modifier' argument types in the dataset. Each of these lexicons will then be independently processed in the next step. This means that arguments such as 'Patient', 'Agent', 'Instrument', 'Benefactor' and others which are usually nouns will not be clustered and condensed. This is because nouns have very few completely synonymous alternatives, and the difference between nouns with similar semantic value may be critical for the particular industry. Thus no vocabulary reduction for nouns takes place.

3. Semantic Clustering of Words in each Predicate/Argument Group

To reduce the size of the vocabulary, the key observation is to realize that a lot of words can be represented as a combination of some fundamental words. For example, gobble can be simply represented as a combination of 'eat' and 'fast'.

This prompted us to form groups of synonyms and choosing one word to represent a whole group, possibly also utilizing combinations to convey the other words in the group. Word embeddings provide a numerical visualization of the semantic interpretation of words. Words with similar semantic value have similar cosine angles. Thus, each lexicon separated in Step 2 is plotted in a multi-dimensional vector space by training embeddings using the enriched GLoVe system as explained in the innovation section. The exact number of dimensions will vary from application to application.

K-means clustering is performed on these vector space embeddings. K-means is intuitively the most appropriate clustering algorithm for our task as we specifically wish to select words that can then be used to represent other words. Clustering on the basis of chosen centres thus makes sense, as this chosen central occurrence forms the cluster representative such that it minimizes command complexity and minimizes loss of meaning, 2 of the 5 optimization parameters of our algorithm as mentioned in Step 4. While derived antonyms have already been taken care of in pre-processing, antonyms with totally different base forms such as those for verbs are treated by enriching GLoVe with troponymy as outlined in the innovation section.

An important question is to choose the 'K' to run K-means clustering. Since estimates of 'K' cannot be made by analysing any properties of the dataset, the elbow method was chosen to determine the optimal K. The human operator can analyze the F-Test (Number of clusters vs Percentage variance) graph and skim over the output clusters to manually correct the choice of K when required as well.

4. Selecting Cluster Representatives

This step forms the crux of our algorithm.

Primarily 5 parameters are accounted for when choosing the representative of each semantic cluster:

a) limiting vocabulary size
b) maximizing phonetic distance between selected words
c) minimizing loss of meaning
d) maximizing familiarity
e) minimizing command complexity

The vocabulary size can automatically be controlled by choosing just 1 representative of each of the semantic clusters. Command complexity depends on the chosen words when the chosen vocabulary is combined to represent different meanings. As a hypothetical example, consider a cluster containing warm, hot and boiling. If boiling is chosen it becomes difficult to represent warm and hot whereas if hot is chosen both other words can be represented accurately. This simultaneously also helps with minimizing loss of meaning. It was thus realized that **c)**, **d)** and **e)** can all be done simultaneously by measuring the euclidean distance of each word in the cluster from the centre of the cluster. The centre of the cluster is computed as the average of coordinates in each dimension over all words of the cluster. Let this euclidean distance for Word **i** be $\mathbf{D_i}$.The familiarity of each word is assumed to be proportional to its frequency in the input dataset after all the modifications in pre-processing have been done. Let this be $\mathbf{F_i}$. These two parameters are independent of the other words chosen for the final vocabulary, unlike b) maximizing phonetic distance between selected words.

Here, the problem is visualized in a different sense. It is important to re-state that we are dealing with one predicate/argument lexicon at a time. The problem of choosing word representatives of different clusters for each predicate/argument is reduced to the Maximum Independent Set problem which is further converted into it's 3SAT equivalent and solved with existing SAT solvers.

For each predicate/argument, a graph is modelled from the current problem. Each word in the lexicon of the predicate/argument is represented as a node in the graph. ABX Testing is performed between each pair of words to check their phonetic similarity using a preset threshold that depends on the application as different environments may have different noise levels which can increase the confusion between word pairs.
Notably, we picked ABX testing over dominating alternatives such as Double Metaphone or Soundex for such a function of identifying phonetic similarities. Soundex was not picked as its 4 digit alphanumeric proved ill suited with gathering a lot of false positives which increases the difficulty of our maximal independent set problem. Examples would be that. We considered expanding on the length of Soundex codes but then realised that the same group of letters '-ough' could be pronounced in multiple ways - 'through' and 'though' - and also often not be flagged with a virtually same sound caused by different groups of letters - 'through' and 'throw'. Double Metaphone was not chosen as while it did provide a comprehensive list of transformations it conducts to come up with phonetic keys for words which can be compared against one another, its utility was limited in our procedure. Other phonetic distance measures like Levenshtein Distance with coefficients learned using hidden markov models [Pucher et al, 2007] were ignored as exact distances need not be known. Our task wasn't to measure how similar two words are phonetically, but rather to prevent two words that are too close to each other from simultaneously appearing in the predicate set. Thus, the technique we decided on was ABX testing.

Given an undirected Graph $\mathbf{G = (V,E)}$ an **independent set** is a subset of nodes $\mathbf{U \subseteq V}$, such that no two nodes in **U** are adjacent. An independent set is **maximal** if no node can be added without violating independence. An independent set of maximum cardinality is called **maximum**. It is evident that choosing a **Maximum Independent Set** of the graph formed ensures that phonetic collisions do not take place. This problem is well known to be NP-Hard and can be reduced to a satisfiability problem by a polynomial time algorithm due to Cook-Levin's Theorem.

However, we must ensure only 1 node per cluster is selected and that $\mathbf{D_i}$ and $\mathbf{F_i}$ as defined earlier are also incorporated. To perform the second task, each node is assigned a weight which is a normalized integral value between 1 and **P,** a parameter that can be varied to ensure optimality based on the application. Let this weight of the node **i** be $\mathbf{W_i}$**.** We now have to maximize the weight collected when an independent set is chosen. An edge is added between all nodes representing words in the same cluster. In other words, for each cluster, each of it's pairs of words are connected with an edge. Now, an independent set can never have 2 nodes from the same cluster as there will always be an edge between them. Thus the remaining problem is equivalent to the **Weighted Maximum Independent Set** problem. To summarize, an edge connects 2 nodes if the words they represent are either in the same semantic cluster or these words are phonetically similar enough to be confused by recognition systems.

Many heuristics for the Weighted Maximum Independent Set problem exist. Some notable approaches are listed. Verhetsel, 2017 uses Indirect Hex-mesh generation. Butenko and Trukhanov, 2006 use critical sets to achieve the same. H Wu use a modified SAT based solver for minimum weighted vertex cover, which can consequently be applied to our problem. [Kako, 2009] extrapolates existing algorithms for the unweighted version of the problem and uses them to approximate solutions for the weighted version. However, the inefficiency of these approaches is evident and it is a well accepted fact that the Weighted Maximum Independent Set problem is understudied.

We take a novel approach by converting the problem back into an unweighted version. The key observation is that since the value of each node is decided by our algorithm itself, it can be normalized to integer values within a set range. This is what is done by choosing a number between 1 and **P** to represent $\mathbf{D_i}$ and $\mathbf{F_i}$. The chosen weight was defined as $\mathbf{W_i}$. Now, $\mathbf{W_i}$ (including original node) unweighted copies of the node are made with the same edge connections. 2 important properties arise which confirm that the problem has been successfully converted into the unweighted **Maximum Independent Set** problem.

Property 1: If a particular node **C** is selected, all its copies can be selected for a better solution.
Proof: Since there is no edge between any of the other selected nodes and **C,** there is no edge between the other selected nodes and the copies. Moreover, since there are no edges within the copies, therefore all copies can be selected thus giving a larger independent set.

Property 2: The weight is accounted for using this process.
Proof: Since Property 1 holds true, selection of a node C means collecting $\mathbf{W_C}$ nodes as all copies get selected.

We chose to go ahead with exponential SAT Solvers because our algorithm is primarily single use and need not be constrained by a tight time bound. Furthermore, since our graph can be observed to be sparse, given that the number of phonetic collisions will be very low and it consists of a large number of small sized clique, it can be assumed that the performance of SAT Solvers will be much better than their worst case complexity. This obviously entails that the chosen limit **P** on the weights is as small as possible. Since accuracy is of higher importance because the vocabulary will then be used in critical industries, the complexity is ignored.

The set selected by the SAT Solver thus gives us the vocabulary of words to be used.

5. Representing whole input dataset sentences in terms of chosen vocabulary

The words which are not chosen in the last step require substitutes for testing purposes. To test the efficiency of our system, the original dataset is mapped

using our proposed vocabulary and then a comparison in speech recognition accuracy takes place. Moreover, it is important to ensure semantic meaning can be wholly preserved. However, this step plays no rĪole in the proposed vocabulary and users will not have to learn the representations produced in this step to convey their meaning. This step is just to illustrate that all sentiments of the user can be conveyed just using the vocabulary that was chosen.

Let the chosen words be a part of the set **V**. Thus, **V'** forms the set of words which are not chosen. For each word **w** in **V,** a WordNet lookup takes place. WordNet proposes a 2-3 word semantic definition for **w**. Let the WordNet definition words be a set **S**. For each word in **S**, the word in **V** with the most similar semantic meaning is chosen. This means, for each $\mathbf{S_i}$, a **j** is chosen such that **Similarity($\mathbf{V_j}$, $\mathbf{S_i}$)** is maximized. The similarity function exists as a part of WordNet. A set **R** is defined such that $\mathbf{R_i}$ is the $\mathbf{V_j}$ most similar to $\mathbf{S_i}$. Each instance of **w** in the input dataset is now replaced with **R**. They are syntactically arranged according to the next step.

6. Establishing syntax for order of predicate verbs, predicate-manner parameters, and arguments to formalize grammar.

In the previous steps we had developed a set for all possible manners of across all verbs in our vocabulary, calling it the set of predicate-manners. Furthermore, we have also sorted our arguments into various headers of the user's specification, in a similar manner. Operating on the assumption of a strictly limited vocabulary, we may infer that whatever expression is come across by the program, it must be validly translated into a sequence of predicates and predicate-manners, taking the arguments as identified from the expression.

In this step we establish a syntactical order for the three components of an expression that we offer, the predicates,predicate-manners, and arguments. We take an example predicate P with predicate-manner set M, argument header set H. Because the predicate performs the key purpose of the expression, it must always be the first word. Thus while designing a syntax, these predicate manners and arguments heads are given the priority and may be recommended alongside the predicates while representing the output. Regardless of whether they appear in the dataset or not however, as long as the users restrain themselves to the predicates, predicate manners, and the argument heads of the vocabulary, their expressions shall be developed to a predicate-argument sequence successfully.

The order for these shall be in the form

Predicate ---> Predicate-Manner ---> Argument Heads.

In order to facilitate easier usage, neither the manners nor the argument heads (nor constituent arguments) are order specific. This reduces the need of users to memorise a certain chain for addressing the system.

Formally, this idea can be expressed in the following manner, borrowing from [Jiang et al.]

Let G{Σ, V, S, F} be a grammar to describe sequences of our program, where:

1. Σ is a finite nonempty set called the terminal alphabet. The elements of Σ are called the terminals.

2. V is a finite nonempty set disjoint from Σ. The elements of V are called the nonterminals or variables.

3. $S \in V$ is a distinguished nonterminal called the start symbol.

4. F is a finite set of functions (or rules) of the form $A \rightarrow B$ where $A \in (\Sigma \cup V)*V(\Sigma \cup V)*$ and $B \in (\Sigma \cup V)*$, i.e. A is a string of terminals and nonterminals containing at least one nonterminal and B is a string of terminals and nonterminals.

One might let the alphabet Σ comprise all words of our vocabulary rather than letters. V would contain nonterminals that correspond to the structural

components in an English sentence, such as <expression>,<predicate>,<predicate-manner>,<predicate-manner phrase> ,<argument phrase> ,<argument-header> ,<argument> , and so on. The start symbol would be <expression>.

Our F would include functions in the form:

<expression> →<predicate><predicate-manner phrase><argument phrase>

<predicate-manner phrase>→<predicate-manner><predicate-manner phrase>

<argument phrase>→<argument-header><argument><argument phrase>

<argument>* →<proper noun>

<predicate>→ <verb>

<predicate-manner>*→ <adverb>

*These functions are not exhaustive to their inputs, and <argument> and <predicate-manner> both can be expanded to include other parts of speech.

7. Training speech recognition model and output of final grammar and vocabulary

The predicates themselves are self explanatory by their design that was created keeping in mind semantic proximity to action. Along with the predicates, the corresponding predicate-manners that occur most frequently are printed as recommended predicate-manners. Similarly, the argument headers that occur most frequently corresponding to the given predicate are printed as recommended argument-headers. It is of course explicitly mentioned that the user may very well use manners or argument headers outside recommended ones, as long as they are members of the vocabulary and can be used in their desired forms within the scope of the rules of grammar established.

The final challenge of the problem is to improve speech recognition accuracy and efficiency by mapping speech to the expressions which are sequences of predicate, predicate-manners, and argument-headers. Speech recognition based on custom defined grammar is a field with a reasonable volume of work done. Currently we will require human operator to manually code in the proposed grammar into custom defined grammar based recognition systems such as CMUSphinx. However, we are looking at potentially directly piping the finalized grammar into a grammar based recognition system thus making a fully integrated end to end system for applications.

Method

While our algorithm is theoretically based on proven assumptions, it will undergo a lot of minor tweaks when applied to different fields. An example that illustrates this is the possibility to use clinical word embeddings [Y Choi et al] for medical applications. A generalized algorithm like ours cannot be proven wholly by success in a single application and thus we will pickup different applications on a case by case basis. These may not always be our target industries, and will be more fundamentally based on availability of the right kind of dataset. A domain where a limited vocabulary system doesn't have applications but for which a textual corpus fitting our needs can be used for case analysis and tweaking the algorithm may also be included among our datasets.

Our program is industry-ended and needs a dataset based on any specific sector that meets a few parameters enumerated below:

We looked for a dataset that has the following key features we were looking for in our datasets -

(i) sentences that are roughly uniform in structure such as commands, prescriptions, consumer complaints (ii) variety in tasks as a small set would not highlight our command generation algorithm sufficiently, (iii) an industry that has a specific vocabulary unique to the industry itself through which can be explore how contextual our program is.

Some of the freely available data sources we picked were legal cases datasets, medical discharge summaries and black box recordings of flights. Tests are to be conducted utilising these datasets to check performance metrics of individual components and then quantitatively calculate the efficiency of the system as whole. Although these datasets don't belong to our target industries they are appropriate for the testing functions. Medical discharge summaries are very rich in properties of specific vocabulary systems whereas black box recordings are a great source of sentences that can have predicate and predicate-manner parallels in context of our program. The findings from their tests may be used to improve on the algorithm minorly as well.

A random selection of 70% of each dataset is taken as *Training Data* (hereafter TD).
In order to increase the reliability of TD, a set of industry specific vocabulary independent from the existing dataset is added separately.

Once all the components are found to be working sufficiently well we integrate the parts together. The predicate set utilises the predicate-manner layers for suitable representation and then each predicate is further detailed with respective argument-headers and arguments. This lays out the output for the first goal of ours. Secondly, an alternate quicker channel is created for speech recognition which tries to map all expressions to a sentence in our formal grammar. This reduces time required to commence action immensely as there lies no ambiguity in the grammar that our program generates, and once successfully mapped, action is immediately undertaken. Furthermore, because of the distinctive commands, errors are much less likely.

While the full pipeline from start to finish could not be integrated, we have either utilized tested work such as K-Means clustering on word embeddings or used both self made and existing input on the novel methods such as sat solvers to test the reduction to 3SAT, paraphrase detection to test if semantic meaning is preserved etc.
In addition to the tests conducted in the individual functions, after the algorithmic engine develops the predicate set, the task ahead was to prove why all of it was necessary at all. This was conducted in an interesting manner. We recognised that the superiority of the suggested procedure is measured using two qualities - noting accuracy change and then checking for any information loss that is crucial using a paraphrase detector. Rest of the dataset as well as the TD (i.e the entire actual dataset available) was processed by a text to speech platform and then back to text. The accuracy was noted. Accuracy for the purposes of our program is defined as the Word Error Rate (WER), a common measure in speech recognition systems that is derived from Levenshtein distances.

Then, the entire dataset was passed through our program, albeit this time attempting to map all sentences to a sequence of the command set we developed as output. This new mapping is obtained and parsed through the previously used Text to Speech converter then back to text using the same speech to text platform, augmented by the formal grammar developed. In our procedure, this program was CMUSphinx. In order to account for varieties that the program could face, we tested it for various voice modulations and accents. Additionally to evaluate how our procedure performed in a robustness test, we used the AURORA database to incorporate noise signals such as recordings from vehicles, street noises, street, airport, restaurant. The method for the same followed from Hu, Y. and Loizou, P. (2007) where a noise segment of the same length as the speech signal was cut out of the noise recordings, scaled and finally added to the machine generated (hence, assumed clean) speech signal.The second phase of our testing process was to run a paraphrase detector on our procedural output and the corresponding input data which would flag any potential loss of crucial

information. These two processes together test our procedure on all fronts ensuring insightful comprehensive results.

Results and Conclusions

Given the industry facing nature of our program we had some trouble procuring a dataset, as there is little publicly available data with command controls, most of the rest being proprietary. We did put in requests for access datasets for research purposes and are expecting responses for the same. We will be utilizing abstract datasets not necessarily made keeping our purpose in mind such as those mentioned in the method section.

We also tried to work out of our handicap by testing the components of the program individually. Our current tests provide generalized results which may be highly misleading as they are hardly exhaustive of all the theoretic points proposed. This is largely because any unattended problems in the pipeline can skew outputs to large measures. Conclusive results are pending and will be available once the individually coded functions are integrated. We will then be able to pick cases and put ourselves in the shoes of the 'human operator', actually optimizing our procedure for various industries.

The parts of our algorithm that were innovative and not result-proven before have been extensively proven in this paper. We have supplemented our reasoning with mathematical proofs where possible and strong intuition where not. We will be able to present comprehensive statistics by the next round.

Given how many technological components have been united together that were never seen in the same place before, their individual efficiencies are also vouched for as the net efficiency of the engine is the product of efficiencies of all components. These components can be used in a multitude of other ways. The cluster optimisation technique that is used for verbs here could also be used for nouns and adjectives or other parts of speech for functions such as paraphrasing to a lower reading level by substituting with function of the optimal words. The method used to identify command qualifiers could be extended to simplifying adjectives. However, the full potential of our procedure will be realized when the need for the human operator is eliminated. To reach that stage, considerable progress in the field of Natural Language Understanding would be required. A more realistic goal would be to integrate a speech recognition system with the procedure so that the final generate vocabulary and grammar automatically govern it's probability models without a human having to explicitly code them in. The development of such a system would then allow what is now a research procedure to convert into a recognition product package that can be distributed to organizations (at a premium), NLP professionals and enthusiasts to apply limited vocabulary recognition to a plethora of tasks which can accommodate a more formal grammar. An open source repository can be setup where people can collaborate to upload and maintain domain specific datasets and pre-trained speech recognition systems for various applications could be an aspirational ideal to hope to achieve by this project. The impact that these results have is therefore more far reaching than merely proving the superiority of our algorithm in the purpose for which it was developed.

Acknowledgement and Reference Links

We would like to the opportunity to thank Anshul Bawa and Dr. Monojit Choudhury of Microsoft Research, India, who guided us at crucial junctures and shed light on a lot of technical complications that we were unaware of previously. Their help motivated us greatly towards our pursuit, and the critical feedback made sure we did not lose our way. Our first introduction to computational linguistics was thanks to Dr. Manish Shrivastava who induced in us a curiosity to seek more about how natural language understanding tasks work.

Our foundations of understanding Natural Language Processing came from the textbook *Speech and Language Processing (3rd Ed.) Jurafsky, Dan and Martin, James H.* We have gathered insights from the following papers for our program:

1. [Johnson, 1999] *Semantic Lexicon for Medical Language Processing,*
2. [Kim, Tur, et. al, 2016)] *Intent Detection Using Semantically Enriched Word Embeddings*
3. [Navaretta, 1999] *Semantic Clustering of Adjectives and Verbs Based on Syntactic Patterns*
4. [Miller et al. (1993)] *5 Papers on Wordnet. CSL report 43, Cognitive Science Laboratory, Princeton University.*
5. [Matuszek 2018] *Grounded Language Learning: Where Robotics and NLP Meet**
6. [Pillai 2018] *Unsupervised Selection of Negative Examples for Grounded Language Learning*
7. [H Yu, 2018] *Interactive grounded language acquisition and generalization in a 2d world*
8. [Durett and Klein, 2013] *Easy victories and uphill battles in coreference resolution.*
9. [Lee et al, 2013] *Deterministic coreference resolution based on entity-centric, precision-ranked rules.*
10. [Kako, 2009] *Approximation algorithms for the weighted independent set problem in sparse graphs*
11. [Kaljurand and Alumae]
12. [C Englund, 2004]
13. [Wei, 2014] *A semantic approach for text clustering using WordNet and lexical chains*
14. [Ferre, 1998] *Voice Command Generation for Teleoperated Robot Systems*
15. [Young, 2017] *Recent Trends in Deep Learning Based Natural Language Processing*

Works Cited

Johnson, S. B. "A Semantic Lexicon for Medical Language Processing." *Journal of the American Medical Informatics Association*, vol. 6, no. 3, Jan. 1999, pp. 205–218., doi:10.1136/jamia.1999.0060205.

Kako, Akihisa, et al. "Approximation Algorithms for the Weighted Independent Set Problem in Sparse Graphs." *Discrete Applied Mathematics*, vol. 157, no. 4, 2009, pp. 617–626., doi:10.1016/j.dam.2008.08.027.