Intent Understanding in a Virtual Agent

Arthi Venkataraman and Ajay Anantha
Wipro Technologies, India
Arthi.venkat@wipro.com, ajay.86@wipro.com

ABSTRACT

This paper discusses the intent recognition system we have built. this system is to be used as part of a virtual agent that can help resolve end user queries. The end user queries are of different intents – request for action, request for information, report of some issue, general greetings. Intent detection is a key component of the virtual agent o decide which type the query belongs to and to further invoke the appropriate action modules. The system uses a combination of machine learning and rules based techniques. The rules based component can be used in an unsupervised mode with only the dictionary databases to be loaded upfront. Classifier is a supervised block which requires training data. The system has a feedback based learning which enables the system's performance to improve with use. This paper brings out the architecture of the intent recognition system, alternate configurations, results obtained and conclusions. The key differentiator of this system is the ability to use this system for different domains with minimal supervision.

CCS Concepts

• **Human-centered computing→Text input**

Keywords

Classifier; Intent, detection; Supervised, learning; Unsupervised, learning; Feedback; Ensemble; Scoring; Virtual Agent; Dependency Graph; Natural Language Processing; Stemming; Tokenization; Machine Learning.

1. INTRODUCTION

In large organizations, there are a lot of queries which are asked by the end consumers. These can include questions of different intents - informational or transactional. Significant effort goes into resolving these queries. There are dedicated people who get into conversation with the end users and help resolve the issue.

We have built a virtual agent system which can get into conversation with end user to understand the problem stated and help resolve it. We have trained system on multiple domains to help solve both internal employee and external customer issue. The challenge with this system is that Virtual agent needs to understand the intent of the natural language statement given by end user. The user can be asking for an information, user may be asking for specific actions, user may be stating a problem he or she is facing or user may be making a generic statement.

The virtual agent needs to be able to distinguish between the different categories of user statements since the next action to be performed will be based on understanding this intent. See Figure 1 for details. There are different back end systems handling different kinds of queries. For example, if the statement is recognized as asking for information the question answering module will need to be invoked to provide response. Similarly different back end systems like action execution module, problem diagnosis module or the generic response module needs to be suitably triggered to give a suitable response. So, from this we see that the intent understanding module is the key module for the running of the virtual agent.

In this paper, we describe the intent detection module. It is built using a combination of natural language processing techniques, machine learning and linguistic rules. The proposed system can be used in an unsupervised manner. Supervision is required in the learning component for adding additional rules or key dictionary terms. This paper discusses the system, its components, performance evaluation criteria, experiments done as well as results.

1.1 Background

We begin by defining the intent recognition task. The input is a sequence of words – *x1* to *xn*. The output is one of the possible values from *y1* to *yk* where *k* is the number of possible output classes. The classes *y1* to *yk* belong to following: Action, Information, Problem or generic. The formulation as above indicates resolution by a text classifier system. However, the challenge of such a system is a need for labelled data. To overcome this, we have followed a multistep technique. Linguistic parsing of the text is done followed by sending the parsed representation to specific linguistic rules. The rules are based on understanding of linguistic patterns in the text. In parallel a separate text classifier is trained using minimal hand labelled data. Any sentence which cannot be understood by the linguistic rules is sent to the classifier system to perform the classification. In the next section, we explain the high level overview of this solution. The detailed solution design is explained in the Solution view section. In the experimental configuration section the results using this system are described.

2. PROPOSED SOLUTION

We are proposing a Virtual agent which can get into conversation with the end user. The virtual agent can understand the end users concerns and help resolve their issues. The virtual agent needs to be intelligent enough to comprehend the query by end user and invoke the appropriate system for processing the request.

While the overall Virtual agent solution has, many modules including Question answering module, action execution module, problem confirmation module, etc. the scope of this paper is restricted to the intent understanding solution to maintain focus.

ICMLC 2017, February 24 - 26, 2017, Singapore, Singapore

ACM 978-1-4503-4817-1/17/02…$15.00
DOI: http://dx.doi.org/10.1145/3055635.3056617

The intent understanding solution uses a combination of a rule based system as well as a machine learning based classification system. The rule based system uses the dependency graph of the input sentence. From this graph, further processing is done to understand polarity of sentence. Check is done to see if the sentence is denoting an action. A further check is done to check if the statement is denoting a question. These values are then fed to the rule module which deduces the intent of the system. In parallel the classifier is run on same sentence to predict the intent.

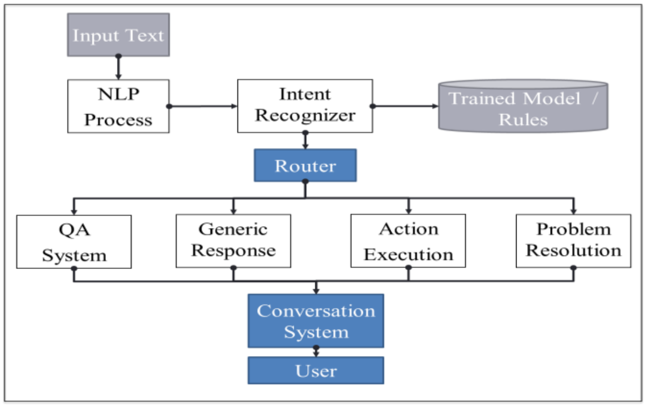

Figure 1. High level view of system

3. PRIOR WORK

The areas of intent detection as well as classifier systems are relevant to our area of work. Few such papers are discussed here along with their relevance to our research.

In [1] multiple classifiers are built by creating random subsets of the features. Resulting classifiers are then ensembled for getting accurate classifications.

In [2] techniques and solutions are discussed to augment a classifier ensemble using automatically generated class level patterns which work more accurately specially for class which have less data.

In [3] multidimensional feature is extracted from the queries. It is then classified using SVM. This is a purely supervised approach as opposed to our approach where we have an unsupervised component which can predict across different domains.

In [4] deep learning based classifiers are discussed.

In [5] techniques are discussed to deduce the intent of the user query using classification techniques. While classification is one of the blocks of our intent detection system we have augmented it with an additional rule based block. The key differentiator in our system being that the designed system will work in an un-supervised manner in any new domain.

In [6] it is shown that we can have large accuracy gains using automatically extracted training data at much lower cost.

In [7] a technique is discussed to use natural language processing techniques like POS tagger and syntactic parser to extract key concepts which are then used to train a support vector machine model to tag semantic units in sentences. We have used a similar feature extraction approach to build the key terms of the different dictionaries. Key difference is that our system has an independent linguistic rules based block which can detect intents in an un-supervised manner.

In [8] techniques and solution for building a large scale accurate classifier is discussed. We leverage a similar technique to build the machine learning classifier block in current paper.

In [9] the relationship between accuracy and number of features is evaluated. This is relevant as we would like to increase the accuracy of our intent detection technique using additional features.

In [10] it has been discovered that use of a wider body of knowledge in the classification process enables a more intelligent and accurate classification.

In [11] large accuracy in training is obtained with the use of automatically extracted bi-grams as part of the classification process.

In [12] it is shown that simple unsupervised TFIDF approach performs reasonably well and the additional information from POS and sentence score helps keyword extraction. We use the TFIDF vectorization as one of the feature generation methods for building our classifier.

In [13] it is shown how an informed feature generation technique based on dependency trees significantly improves clustering quality. As part of our intent detection approach, dependency trees form an important input for extracting features for the rule based system.

4. SOLUTION VIEW

Figure1 depicts a high-level view of the virtual agent system.

Figure 2 gives a detailed view of the Intent module.

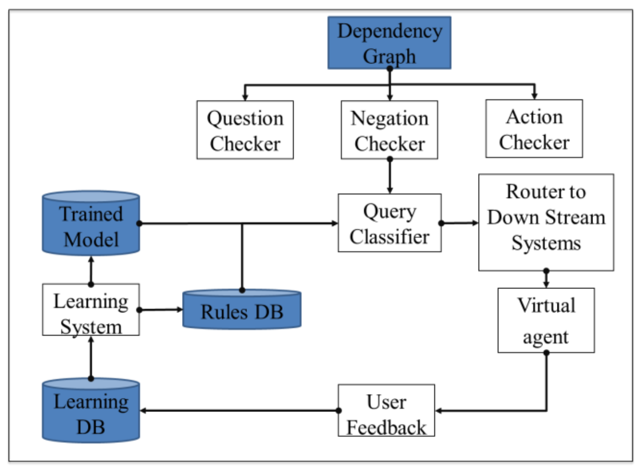

Figure 2. Detailed view of Intent module

The key blocks of the intent identification system are described below:

4.1 Dependency Graph Extractor

This module extracts the dependency graph of the sentence. The key relevant output of this module is the Action verb with the corresponding noun.

4.2 Question Checker Module

This module identifies if the given sentence is a question. It uses the trained question classifier model which classifies the sentence into a question. For example, “How do I install a printer?” will be classified as question.

4.3 Negation Checker Module

This module checks if the given sentence is a negative sentence. It uses the negation dictionary to do this classification. This dictionary is domain specific. The negation dictionary is configurable. Some of the terms in the dictionary are Don't, Unable Cant, Crashed, etc. A simple keyword presence match is currently followed for this. Post tokenization, stemming and lemmatization a check is done for presence of the negative terms. Let us consider the example – "My PC crashed" and "I'm unable to print". Both above sentences are recognized as negative because of presence of crashed and unable in the two sentences respectively.

4.4 Action Checker Module

This module checks if the given sentence has a relevant action verb. It uses the action dictionary to do this classification. This dictionary is domain specific. The action dictionary is configurable. Some of the terms in the dictionary are install, setup, configure, fix, check, etc. The main and supporting action verbs are got as part of the output from the dependency graph. A simple keyword presence match is followed for this. Post tokenization, stemming and lemmatization a check is done for presence of the action terms in the action term dictionary. Let us consider the example – "I want to install a printer". The above sentence is recognized as containing action because of presence of install in sentence.

4.5 Query Classifier Module

This module classifies the given sentence into the correct intent category with classification being provided by both the Rules based and classification system.

4.5.1 Rules Based Intent Recognition

This module refers to the Rules dictionary and takes as input the output of the three different checker modules. It then predicts the intent class of the given sentence. Rules are configurable. The rules are developed based on deep understanding of the domain and user interactions. For example, it is observed a statement with only a negative flag enabled is usually a problem. Similarly, a statement with only the action flag is an action. A statement with both a question and action components is recognized as asking for Information. A sample view of the rules is given below.

Table 1: Rules matrix

ID	Question	Negation	Action	Type	E.G
1	0	0	0	General	Hello, Hi
2	0	0	1	Action	I want to install a printer
3	0	1	0	Problem	My printer crashed
4	0	1	1	Problem	I'm not able to install my printer
5	1	0	0	Information	Where is the software icon?
6	1	0	1	Information	How do I install a printer?
7	1	1	0	Information	Where is the machine that isn't working?
8	1	1	1	Action	Can you fix the printer that is crashed?

Note the rules are configurable. More rules can be added. Also, based on domain certain rules will be modified for a different behavior.

4.5.2 Machine Learning Based Classifier for Intent Recognition

This module uses a supervised discriminative approach (Ridge classifier) to classify a sentence into an appropriate intent category. The training data for this is a set of labelled sentences for each of the four categories. Figure 3 is the block diagram of the classification system. Note the classification block diagram is borrowed from Arthi [2].

The natural language sentence is in a higher dimension space. In the NLP Process block the unstructured text in tokenized and the important features extracted. The features are then converted to TF-IDF vectors and compressed to reduce the dimensionality.

These features are fed to the ridge classifier. The ridge algorithm learns a linear plane which uses maximal marginal separation to separate the different data points into separate classes. The trained model is stored in the trained model store.

When a new sentence comes in it is suitably processed using natural language processing techniques and sent to the predictor to predict. The prediction block uses to trained model and predicts to which intent bucket the given sentence falls in.

There is feedback loop which accepts negative feedback from end users. When users of the system give negative feedback, agents can go and manually label such cases with the correct intent. The models are regularly re-trained using this data. This is explained in the Learning section.

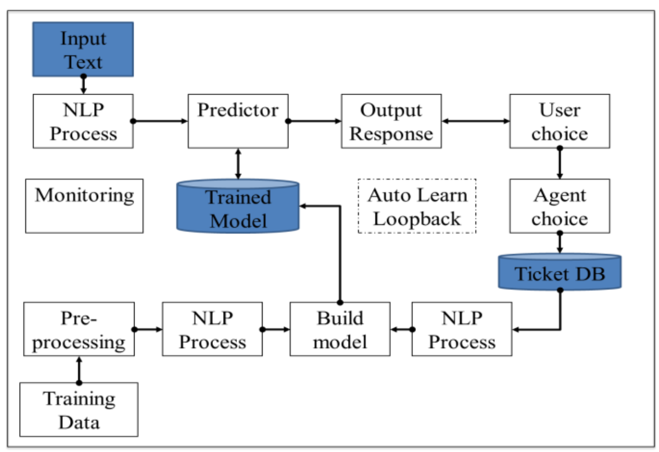

Figure 3. Detailed view of classifier.

5. LEARNING

This module helps in learning so that system performance improves with use. It refers to learning database where the feedback from user on intents is captured. In all cases where system did not do well additional rules are added to make the system better. The classifier is also re-trained with the sentence and correct intent class.

6. EXPERIMENTAL SETUP

6.1 Experimental Configurations

We used multiple configurations for intent predictions in our approach to this experiment. The different configurations are tested with the data set and the output will be measured using defined metrics. A comparison will then be arrived at for the performance across different configurations.

6.1.1 Configuration 1 – Rule Based Approach for Intent Detection.

In this case training and prediction is performed using only the rules based module as described in section 4.5.1.

6.1.2 Configuration 2 – Classifier Approach for Intent Detection

In this case intent classification is performed by only the classifier system as described in section 4.5.2. The classifier system is trained with a small corpus of hand tagged sentences. The corpus for training was small since it is a time consuming and manual process.

6.1.3 Configuration 3 – Combination of Rule based and Classifier approach

In this setup, the predictions from Configuration 1 was used to generate labelled data. The rule based approach predicts well in the scope of its rule. Beyond the scope of the rules it predicts “I don’t know” responses. This feature enabled us to generate more training data. This training data was used to train the classifier as stated in 4.5.2. Hence while technically the setup is same as Configuration 2 the key difference is that the model in this configuration is trained with more data from the rules based system.

6.2 Data Set

6.2.1 Training Data

For training the classifier algorithm we used two data sets as below:

- *Data Set 1*: A smaller manually labelled data set with 50 query sentences.
- *Data Set 2*: A larger data set generated through predictions from Configuration 1 having 300 query sentences.

Table 2: Data composition of Training Set

Class Name	***Number of Data Points for Data Set - 1***	***Number of Data Points for Data Set - 2***
Information	12.00	50.00
Action	18.00	90.00
Problem	18.00	140.00
Generic	2.00	20.00
TOTAL	**50**	**300**

6.2.2 Testing Data

A set of 160 query sentences having a mix of all the above classes was used to predict. The testing data was common across all the three configurations.

6.2.3 Sample Data

Below are examples of queries for each class name:

- Action (ex: I want to install skype.)
- Information (ex: How to install skype on windows?)
- Problem (ex: I am unable to install skype.)
- Generic (ex: What is the weather like?)

6.3 Evaluation Metrics

The overall system built needs to be able to choose the correct intent from all supported intent classes. This is similar in functionality to a classifier. F-score (Harmonic mean of precision and recall) has been chosen as a metric as it is able to correctly measure the combined effect of precision (which tells us what proportion of events system identified as having an intent actually is of that intent) and recall (which tells us what proportion of intents given by system that actually had an intent were diagnosed by system as having the intent).

7. RESULTS

The Intent prediction from all the three configurations were analyzed and performance measures for each were calculated. The predictions from configuration 1 is a dynamic process as the rule is applied on every sentence that it is tested on. It showed a high recall but lower precision. This could be due to the rules being restricted to the domain specific action verb and negation word dictionaries. While predictions from configuration 2 is relative to the training dataset that was used. It had good precision and f-score and proved to be a relatively better model.

The results from the above two configurations led us to configuration 3. As the Configuration 1 generated few false negatives, the predictions could thus help us build a labelled dataset for the classification system. With this, we could overcome the limitations of configuration 2 and expand the prediction capability of the overall model. The resulting configuration as expected had a similar recall but higher precision as compared to the configuration 2 while handling wider variety of sentence.

Table 3: Confusion matrix for Configuration 3

	Predicted-NO	Predicted-YES
Actual-NO	25	9
Actual-YES	14	112

The below figures highlight the performance measure parameters across the three configurations.

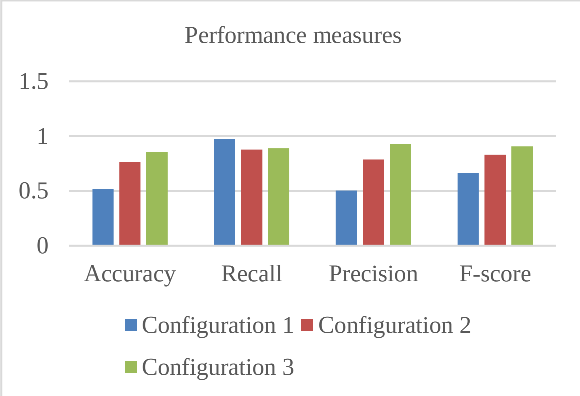

Figure 4. Performance measure scores.

8. CONCLUSIONS

Understanding the intent of a query is an important aspect of automated resolution through a virtual agent. Intent classes enable to decide the next action to be made by the system. The strength of rule based intent recognition is in getting higher number of correct predictions with restricted scope of domain specific knowledge dictionary, while that of a machine learning based classifier is to handle variety of sentences for predictions. Though the rule based approach covers closed domain problems and handle complex query system representing multiple intents in each sentence, the scope of this configuration is also strictly restricted to the rules. However, the queries outside this would be effective feedbacks for learning the new rules. The classifier on the other hand lacks labelled data for training the different intents classes. The combination of these two methods enabled the effective application of this system in giving a strong model which gave good number of correct predictions along with broadening the scope of prediction.

9. WAY FORWARD

We would like to experiment on Deep learning based classifier for building the classifier systems. We would like to automatically learn the dictionary terms for the action and negative phrase dictionaries using unsupervised knowledge extraction techniques. We would like to experiment on learning the rules for classification in an automated manner. One more area is to build classification systems with additional features like the action verbs, presence and absence of negative terms, query phrases, etc.

10. ACKNOWLEDGEMENTS

We give our heartfelt thanks to our organization Wipro which has enabled us to build this system. They have also provided the needed support to make this presentation.

11. REFERENCES

[1] Attribute bagging: *improving accuracy of classifier ensembles by using random feature subsets by Robert Bryll, Ricardo Gutierrez Osuna*, Francis Quek in Pattern Recognition Volume 36, Issue 6, June 2003, Pages 1291–1302

[2] *Augmenting a Classifier Ensemble with Automatically generated class level patterns for Higher Accuracy*, Arthi Venkataraman, 2015 IEEE International Conference on Technologies and Application of Artificial Intelligence held in Taiwan

[3] *Automatic Identification of Query Intent by Using Multiple Features*. JIANG Zong-li,ZHANG Heng-qing. School of Computer Science, Beijing University of Technology. Published in Computer and Information Technology , 1005-1228, 2015 Issue 1

[4] *Deep Learning Algorithms based text classifier. Arthi Venkataraman*, 2016 IEEE International Conference on Applied and Theoretical Computing and Computation Technology held in Bangalore

[5] *Determining the User Intent of Web Search Engine Queries*. Bernard J. Jansen, Danielle L. Booth, Amanda Spink. WWW 2007, May 8–12, 2007, Banff, Alberta, Canada. ACM 978-1-59593-654-7/07/0005.

[6] Improving Classification Accuracy Using Automatically Extracted Training Data by Ariel Fuxman, Anitha Kannan and others. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining Pages* 1145-1154

[7] Learning from syntax generalizations for automatic semantic annotation Guido Boella ,Luigi Di Caro ,Alice Ruggeri , Livio Robal Published online: 27 May 2014 © *Springer Science+Business Media New York* 2014

[8] Machine learning techniques for building a large scale production ready classifier by Arthi Venkataraman. *Proceedings of the Second International Conference on Data Mining, Internet Computing, and Big Data, Reduit, Mauritius* 2015

[9] *On the Relationship between Feature Selection and Classification Accuracy Andreas G. K. Janecek andreas*.janecek@univie.ac.at Wilfried N. Gansterer wilfried.gansterer@univie.ac.at, FSDM, 2008

[10] Overcoming the Brittleness Bottleneck using Wikipedia: *Enhancing Text Categorization with Encyclopedic Knowledge by Evgeniy Gabrilovich and Shaul Markovitch Department of Computer Science Technion*—Israel Institute of Technology, 32000 Haifa, Israel {gabr, shaulm}@cs.technion.ac.il (Unpublished)

[11] *The use of bigrams to enhance text categorization* by Chade-Meng Tan,Yuan-Fang Wanga,,Chan-Do Lee Information processing & management, 2002

[12] Unsupervised approaches for automatic keyword extraction using meeting transcripts. By Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. In *Proceedings of Human Language Technologies*: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL '09, pages 620–628, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[13] *Unsupervised Discovery of Relations and Discriminative Extraction Patterns*. Alan Akbik, Larysa Visengeriyeva, Priska Herger, Holmer Hemsen, Alexander Löser. Proceedings of COLING 2012: Technical Papers, pages 17–32, COLING 2012, Mumbai, December 2012