Identification of Unstructured Language Indicating Multiple Objects

by

Vivek Paramasivam

Supervised by Luke Zettlemoyer

A senior thesis submitted in partial fulfillment of

the requirements for the degree of

Bachelor of Science
With Departmental Honors

Computer Science & Engineering

University of Washington

June 2015

Presentation of work given on ____Nov 5, 2014_____

Thesis and presentation approved by _____

Date May 7, 2015

Abstract

We present a natural language classifier that demonstrates accurate differentiation between sentences which refer to a single object and sentences which refer to multiple objects. This classifier gives 90.25% accuracy on our corpus. We describe the features which went into the design of the classifier, and show that it is both effective and is a useful tool in object identification tasks.

Contents

Introduction

Object identification tasks are prevalent in many contexts in robotics. Particularly, there has been a push towards unstructured interaction with machines in order to allow untrained users to interact with robots. A portion of this problem involves parsing natural language to derive the intent of the user. In the case of object identification tasks, the problem can be broken down further into determining how many objects the user is referring to, and then identifying a proper subset of that size from a given set of objects to choose from.

One of the most common and clearest distinctions is between sentences which refer to single and multiple objects, due to the differences between the ways we refer to singular and plural numbers of objects in the English language. If we know how many objects are being referred to by spoken language, we can implement appropriate logic to improve our identification results.

Contribution

In this paper, we take a step towards improving natural language based object-identification algorithms by developing a classifier which accurately determines whether a spoken phrase refers to one object, or refers to multiple objects, and describe scenarios in which this classifier will find use.

Problem Definition

Our goal is to develop a classifier which, learned on features derived from a corpus of data and given a new sentence or phrase, can determine whether the speaker of the sentence was referring to one object or to multiple objects.

We assume data given as sentences or phrases in English in which the language refers to at least one object, but make no assumptions about the syntax, word choice, or sentence structure of the data. The goal is to make this classifier usable in as many contexts as possible.

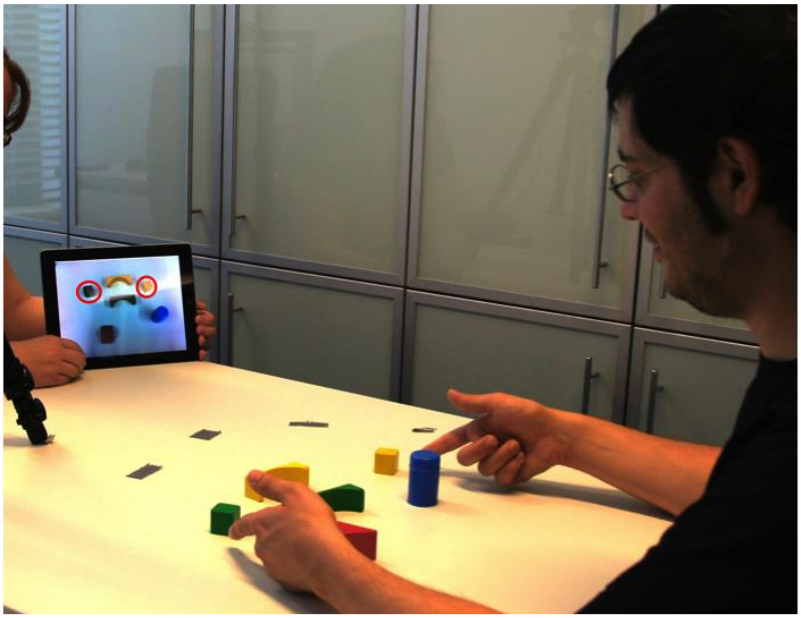


Fig. 1: Data collection. (Matuszek et. al.)

Data Collection and Corpus

The corpus used for learning was collected by transcribing the language of individuals indicating specific objects in a scene (Matuszek et. al.). Participants in this data collection were asked to indicate objects using both language and gesture, and were told which objects to indicate through an image of the scene with the correct objects circled (see Fig. 1). The participant's gestures were recorded by a tripod-mounted Kinect sensor, and their spoken words were transcribed by hand.

In collecting this data, 28 different scenes were used, each varying the number and locations of the objects on the table, as well as varying which objects were to be indicated. The number of objects the participant was asked to indicate varied from one object to four objects. There were a total of 13 participants describing each scene, but some participants did not use spoken language to indicate objects, choosing to use only gesture. Those instances have been removed from the data set for the purposes of this investigation, leaving 358 instances to analyze for our purposes. Some participants described scenes in as quickly as three seconds while others took as long as a minute.

The relevant data from this dataset for this investigation are the transcribed phrases the participants uttered to describe each scene, and the number of objects each participant was meant to indicate. Thus, we have a corpus of data of the form:

<*sentence*, *numIndicated {1, 2, 3, 4}* >

For example:

<”The red block and that blue one”, 2>

<”This one right here”, 1>

<”These three”, 3>

An important distinction to make here is that we do not manually analyze each sentence to parse the number of objects the language indicates to find ground truth. Rather, we simply use the number of objects the user was requested to indicate as the ground truth value of *numIndicated*.

However, in this particular investigation, the relevant information is if the user was intended to indicate a single object, or was supposed to indicate multiple objects. In order to achieve this, we convert *numIndicated* to a binary value *multIndicated*, which takes the value 0 if exactly one object was meant to be indicated in that scene, and takes the value 1 if multiple objects were meant to be indicated. Our final input dataset then consists of entries which take the form:

<*sentence*, *multIndicated {0, 1}*>

Modifying our previous examples:

<”The red block and that blue one”, 1>

<”This one right here”, 0>

<”These three”, 1>

Previous Work

This project adds on to the work done by Matuszek et. al. (2014) in understanding unstructured language. Their work analyzed a combination of gestural and verbal communication to identify a subset of indicated objects from a scene of objects. The goal going into this project was for our classifier to improve the results of their system. Thus, we use their corpus to perform our learning and analysis.

A crucial feature of our classifier is the inclusion of part-of-speech occurrences in the input dataset. Toutanova et. al. developed a part-of-speech tagging utility (2003) which gives 97.24% accuracy on the Penn Treebank WSJ tagset and we use the results of their work to perform part-of-speech tagging of data for our classifier.

Another critical feature we extract from the data is the occurrence and frequency of linguistic phrases. For this purpose we use The Stanford Parser (Klein et. al. 2003; Socher et. al, 2013)

Classifier

All classification in this study is done through logistic regression. Specifically, we used the standard implementation of SimpleLogistic in the WEKA machine learning toolkit.

Feature Selection

Recall that our input dataset is of the form:

<*sentence*, *multIndicated {0, 1}*>

In order to perform logistic regression on this dataset, it must be converted to a series of feature vectors of the form:

<{features}, multIndicated {0, 1}>

Where {*features*} is a set of features which describe a particular sentence in the dataset and *multIndicated* is, as before, the ground truth as to whether or not that sentence was meant to refer to a single object (0) or multiple objects (1).

Each spoken sentence in the corpus had to be converted into such a feature vector and the selection of appropriate features was a key part of developing the classifier. We approached this problem iteratively, starting with simple features such as line length, then adding and refining features to steadily improve accuracy. Ultimately,

we arrived at a classifier which uses three main types of features, which we refer to as Bag of Words, Parts of Speech, and Phrases.

Bag of Words

For this set of features, we created a bag-of-words feature vector from *W,* the set of all words found in the corpus, for each sentence of data in the corpus. In this feature vector, each word *w* takes a boolean value {0, 1} representing its presence or absence in the particular sentence.

For example, if our corpus consisted solely of three sentences as follows:

< "The blue ones", 1>

< "The green one in the middle", 0>

< "This blue one" , 0>

Then the set of all words in our dataset is:

{"the", "blue" , "ones", "green", "one", "in", "middle", "this"}

Now, each of the entries in the corpus can be represented as:

< "The blue ones", 1> → *{1, 1, 1, 0, 0, 0, 0, 0, 1}*

< "The green one in the middle", 0> → *{1, 0, 0, 1, 1, 1, 1, 0, 0}*

< "This blue one", 0> → *{0, 1, 0, 0, 1, 0, 0, 1, 0}*

In addition, we included the number of words in each sentence, *lineLength*, in each feature vector:

< "The blue ones" , 1> → *{1, 1, 1, 0, 0, 0, 0, 0, 3, 1}*

< "The green one in the middle" , 0> → *{1, 0, 0, 1, 1, 1, 1, 0, 6, 0}*

< "This blue one" , 0> → *{0, 1, 0, 0, 1, 0, 0, 1, 3, 0}*

Parts of Speech

Each token in a spoken sentence has a logical part of speech associated with it. We used the Stanford NLP tool to perform our Part of Speech tagging, which uses parts

of speech as defined by the Penn Treebank Tagset (Santoniri, 1990). A sample of these parts of speech are listed in Table 1.

Table 1 – Sample parts of speech (Penn Treebank Tagset)

CC	Coordinating conjunction e.g. and, but, or
CD	Cardinal Number e.g. one, two, ten, one hundred and thirty
DT	Determiner e.g. The, a, my, this
JJ	Adjective
JJR	Comparative Adjective e.g. more, less, taller, faster
JJS	Superlative Adjective e.g. least, tallest, shortest
NN	Singular Noun
NNS	Plural Noun
PRP	Personal Pronoun e.g. I, he, you
PRP$	Possessive Pronoun e.g. my, his, yours
RB	Adverb e.g. quickly, slowly
VB	Verb

We would imagine that the presence or absence of certain parts of speech, in a sentence, such as NNS (plural noun) or CD (cardinal number), may be telling of the number of objects referred to in that sentence.

With this in mind, we created a set of features which represent the number of occurrences each of these parts of speech in a sentence, in a similar manner to the way we generate the Bag of Words. The notable difference here is that while the Bag of

Words was entirely boolean values, our Parts of Speech features can take the value of any non-negative integer. At this point, our feature vector consists of:

<{Bag of Words}, {Parts of Speech}, lineLength, multIndicated>

Phrases

The final set of features in our feature vector are linguistic phrases which are effectively multi-word parts of speech, which we recognize through the use of the Stanford NLP Parser tool. Examples of these phrases are listed in Table 2.

Table 2 – Sample Phrases

PHRASE	**Description**	***Example***
NP	noun phrase	*The red block*
PP	prepositional phrase	*in front of*
VP	verb phrase	*is sitting*
ADVP	adverb phrase	*quickly*
ADJP	adjective phrase	*very blue*

Just as with the Parts of Speech, the number of occurrences of each “phrase” in the sentence is defined as a feature in the feature vector. Thus, our final feature vector consists of:

< {Bag of Words}, {Parts of Speech}, {Phrases}, lineLength, multIndicated >

The input to the logistic regression classifier consists of 358 such feature vectors, each representing a different spoken sentence from the original dataset.

Evaluation

We used two evaluation metrics to analyze the success and effectiveness of the classifier. First, we assessed how well it performing the task it was designed for by comparing it to a human baseline on the same task. When we were satisfied by its effectiveness, we gauged its usefulness by incorporating it into a larger system.

Effectiveness compared to human baseline

In order to establish a human baseline, we assumed that an average English-fluent human would be reasonably be able to determine if a specific sentence from our corpus referred to a single object or to multiple objects so long as the sentence does not contradict itself, and does not directly contradict the ground truth.

For example, one of the descriptions of a scene was:

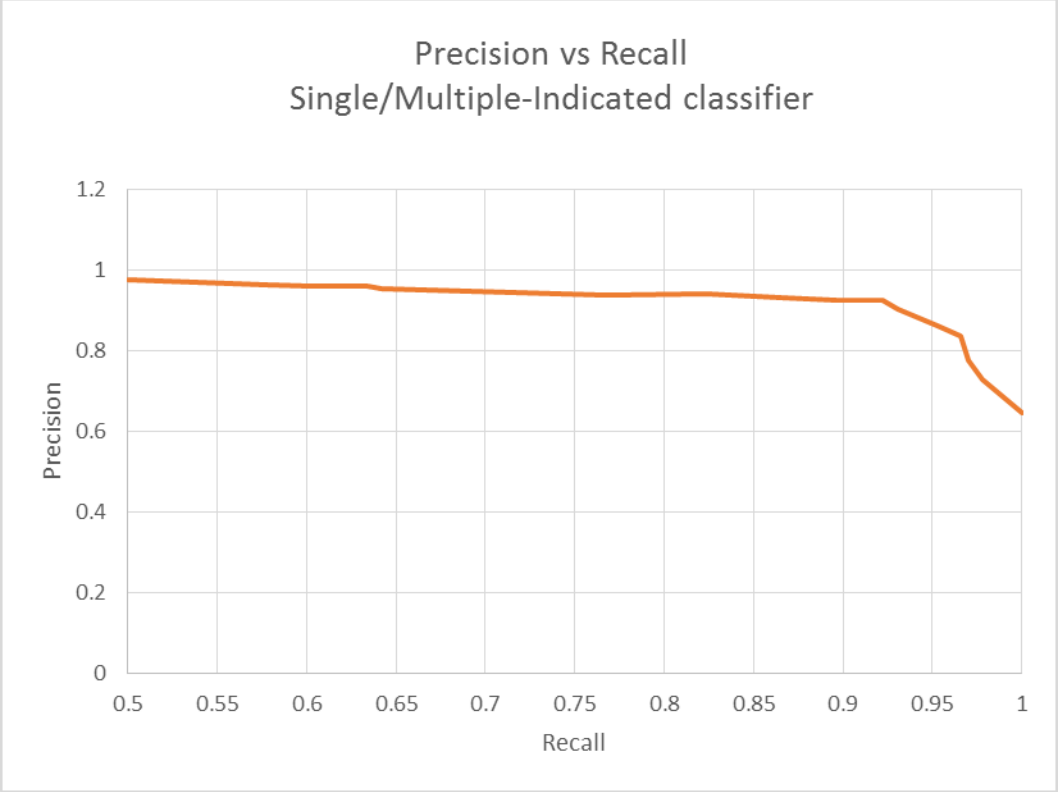
"The middle one"

The speaker of this sentence, during our data collection as described in the Data Collection and Corpus section, was actually asked to refer to two objects, but the language implies that only one object is being referred to. It directly contradicts the ground truth, so it would not be surprising for someone who is shown this sentence to evaluate it incorrectly. We would not expect our classifier to be able to assess this accurately either.

Manually analyzing our dataset, we found that it contained nine such contradictory sentences. These nine instances similar to the one described above make up 2.5% of the dataset, leaving 97.5% of the data reasonably able to be understood by a human. We thus consider 97.5% to be the gold standard for accuracy which we hope for our classifier to achieve.

Using logistic regression on the dataset of feature vectors described in the previous section, using a naïve cutoff of 0.5, the classifier achieved 90.3% accuracy, correctly labelling 324 of the 358 instances. It identified descriptions which indicated multiple objects with precision 0.926 and recall 0.922. It turns out that this threshold of 0.5 gives us close to the highest possible recall (0.922) without significantly losing precision, as can be seen in Fig. 2, which was produced by varying the threshold.

Fig. 2 – Precision-Recall curve for Single/Multiple-Indicated classifier



The achieved 90.3% accuracy was 7.2% below the human baseline, satisfactory enough to move on to testing it in a larger system.

Evaluation by implementation

We used our classifier in the context of data again provided by Matuszek et. al – a system consisting of a robotic arm and a Kinect camera, in which a user indicates objects on a table through gesture and language. For each object on the table, the system evaluates whether or not the object was indicated, then uses the arm to pick up those objects and move them into a box. We used the multIndicated classifier to help identify whether or not an object was indicated.

The system designed by Matuszek et. al. identified indicated objects by calculating two values for each object in the scene, *vizlangprobability* and *gestureprobability*, which, respectively, are the probability that the user's spoken language indicated an object and the probability that the user's hand gestures indicated

an object. The initial dataset consisted of 520 feature vectors, each of which consisted of those two values, each representing a single object from some scene. Using the standard implementation of logistic regression in the WEKA machine learning toolkit, and an optimzed indicated-cutoff of 0.31, the object-indicated classifier was able to correctly identify whether or not an object had been indicated 88.65% of the time. We used that value of 88.65% as the value to improve through the use of our multIndicated classifier.

We tried two approaches in integrating our classifier into this system. The first, naïve method was simply to include, for each of the 520 feature vectors, the *multIndicated* probability value for the scene which contained the object that the feature vector represents. That is, each of the feature vectors now had a third value which was a probability, between 0 and 1, that there were multple objects indicated in the particular scene from which the object was extracted. The idea was that if we knew that multiple objects were indicated in a scene, there would be a slightly higher chance that any particular object in that scene was indicated. We hoped that in scenes with high *multIndicated* probabilities, this slight increase in probability would cause some objects to be pushed over the threshhold for indication and be properly counted as being indicated. However, the increase in probability proved insufficient for this purpose.

With the *multIndicated* scores added to each vector, using logistic regression, the system was able to accurately identify whether or not 88.89%of the time – a negligible difference from the 88.65% accuracy the system had prior.

As described in the introduction, the true power of this classifer is to be able to perform logical operations on its output to change the way that data is classified on a scene-by-scene basis, which is how we designed the second approach.

In the second approach, we performed logistic regression on the dataset as described in the first approach in order to calculate the probability that each object was indicated, then wrote a script to analyze the data scene-by-scene. First, we mapped each object in the dataset to the scene from which it came. Then, for each scene, we iterate through each object and use the object's calculated *probIndicated* output value from the logistic regression to determine whether or not it was indicated. After analyzing a scene in this way, we count the number of indicated objects in that scene, and check

to see if it agrees with the value of *multIndicated* for that scene. If *multIndicated* tells us that a single object was referred to in the language, but the existing system returned more than one object indicated in the scene, we instead only mark the object in the scene with the highest *probIndicated* value as indicated. If *multIndicated* tells us that multiple objects were indicated in a scene, but the existing system returned one or zero indicated objects, we set more objects as being indicated until we have two indicated objects in the scene. In other words, we validate that each scene agrees with the corresponding value of *multIndicated*, and make changes where there are disagreements, siding in the favor of the output of our *multIndicated* classifier.

As mentioned above, an object was initially considiered indicated if its *probIndicated* score, the output from the object-indicated classifier, was above a certain threshhold. Similarly, a scene was considered to have multiple objects indicated if its *multIndicated* value was above a certain threshhold. We varied both of these values from 0.0 to 1.0 in increments of 0.01, testing every combination, resulting in 10,000 tests run. We found our optimal result, 90.5% accuracy, with a *probIndicated* cutoff of 0.54 and a *multIndicated* cutoff of 0.37.

We had hoped to see a 2.5% improvement in accuracy, and the result of 90.5% was only a 1.85% improvement over the 88.65% accuracy we saw before the inclusion of our classifier. Still, the improvement was significant enough to show that by adding this logical verification based on the *multIndicated* classifier we developed, the results of object identification from natural language can be improved.

Extension

The next step from here would be to add on to the *multIndicated* classifier to turn it into a *numIndicated* classifier, which would determine how many objects were referred to in the language exactly, and then perform similar logic as described in the previous section. In our preliminary tests of such a classifier using similar features, we found that achieving a high accuracy on such a metric was not as simple as it was with the *multIndicated* classifier, namely because the difference between single and multiple is much more pronounced in the grammar of the English language then between, for example, three and four objects being referred to.

Ultimately, we hope that this classifier will be a basis and building block for more advanced language-analysis classifiers which can further enhance object identification tasks, such as a system which could identify the number of objects referred to in language which match a certain set of criteria. For example, a classifier which could answer questions such as “How many objects did the speaker refer to as red in this sentence?” could be very helpful in verifying the output of visual and gesture-based identification systems.

Acknowledgements

Many thanks to Luke Zettlemoyer for helping to scope out this investigation and discussing ideas with me, and to Cynthia Matuszek for her fantastic mentorship, advice, and support.

References

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423-430.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL 2003*, pp. 252-259.

Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. 2014. Towards Understanding Unconstrained Gesture and Language Input for Human-Robot Interactions. *Submitted to the 2014 IEEE International Conference on Robotics and Automation (ICRA).*

Richard Socher, John Bauer, Christopher D. Manning and Andrew Y. Ng. 2013. Parsing With Compositional Vector Grammars. *Proceedings of ACL 2013*

Santorini, B. 1990. Part-of-speech tagging guidelines for the Penn Treebank Project. Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.