Paraphrasing Invariance Coefficient: Measuring Para-Query Invariance of Search Engines

Tomasz Imielinski
Ask.com
Tomasz.Imielinski@ask.com

Jinyun Yan
Rutgers University
jinyuny@cs.rutgers.edu

Yihan Fang
Ask.com
Yihan.Fang@ask.com

Kurt Eldridge
Ask.com
Kurt.Eldridge@ask.com

Huiwen Yu
Ask.com
Huiwen.Yu@ask.com

Peter Kelly
Ask.com
Peter.Kelly@ask.com

ABSTRACT

Paraphrasing is the restatement (or reuse) of text which preserves its meaning in another form. A para-query is a paraphrase of a search query. Humans easily recognize para-queries, but search engines are still far away from it. We claim that in order for a search engine to be called semantic it is necessary that it recognizes para-queries by returning the same search results for all para-queries of a given query. Recognizing para-queries is an important and desired ability of a search engine. It can relieve users of the burden of rephrasing queries in order to improve the relevance of results.

In this paper, we cover two main threads: monolingual para-query generation (PG) and para-query recognition measurement (PRM). Para-query generation aims to automatically generate as many English para-queries as possible for a given query. We propose a novel game "Rephraser" to tackle this problem. Hundreds of para-query templates are extracted from the game's output and used to compose tens of thousands of para-queries.

The goal of para-query recognition measurement is to examine to what level search engines recognize para-queries. We propose the concept of paraphrasing invariance coefficient (PIC) which is defined as the probability that search results are the same for a pair of para-queries. By using para-queries generated from the game, we design experiments to measure search engines' PIC. Results show that today's leading search engines are still inferior to human ability in recognizing para-queries. It is a long way ahead for search to be truly semantic.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Semantic Search, Search Engine Evaluation and Measurement, Natural Language, Query Diversity

General Terms

Measurement, Design, Experimentation

WWW2010, April 26-30, 2010, Raleigh, North Carolina.
.

Keywords

Para-query generation, Para-query recognition measurement, Rephraser Game, Paraphrasing invariance coefficient, Semantic Search Engine

1. INTRODUCTION

In[7] we introduced the notion of semantic invariance. We stated that for a truly semantic search engine it is necessary (not sufficient) to preserve semantic equivalence of queries. Semantically equivalent queries should yield the same results, just like humans would. We define para-queries as semantically equivalent queries in the paper. In fact semantic equivalence is one of the "you know it when you see it" features. For example, we have these para-queries:

1. arthritis cause
2. what causes me to get arthritis
3. what gives me an arthritis
4. why do i get arthritis
5. why do i have an arthritis
6. why do people get arthritis
7. why arthritis
8. what causes an arthritis

In response to above queries, a human would give the same answer which would be the best web page to describe the causes of arthritis. Paraphrasing the query should not change the decision about which page is the best source of information for "causes of arthritis". In fact, regardless of which paraphrases above is used, the same URL should be selected by semantic search engines as the top result. Unfortunately it is not the case today. We sent above queries to today's leading search engines. Table 1 shows returned top URLs by Google. Only query No. 1 "what causes an arthritis" and query No. 8 "arthritis cause" have same top returned URL. Other queries produced different results.

Recognizing para-queries is an important requirement for semantic search engines. If search engines could learn to recognize para-queries, users would not need to reformulate their initial queries to get better results. Any para-query of a given query would yield the same results. The burden of understanding para-queries would fall on the search engine,

1	http://orthopedics.about.com/od/hipkneearthritis/f/arthritis.htm
2	http://www.livingwithrheumatoidarthritis.com
3	http://www.arthritistoday.org/conditions/rheumatoid-arthritis/all-about-ra/diagnosing-ra.php
4	http://arthritis.about.com/cs/jra/a/childrentoo.htm
5	http://www.niams.nih.gov/Health_Info/Arthritis/tengo_artritis.asp
6	http://www.infinitehealthresources.com/Store/Resource/Article/1-45/2/960.html
7	http://www.acefitness.org/media/media_display.aspx?itemid=187
8	http://orthopedics.about.com/od/hipkneearthritis/f/arthritis.htm

Table 1: Returned Top Results from Google

not on the user.

Given two para-queries, our goal is to test how likely it is that a search engine would return the same result. To reveal the answer, we need a massive number of para-query pairs. Unfortunately there are no effective methods to determine whether two queries are semantically equivalent, nor is there a universal method to generate all para-queries for a given query. In this paper we used three methods for para-query generation:

1. synonym replacement
2. query augmentation
3. Rephraser game

As to method (1) and (2), we have used them in [7] where we generated para-queries by replacing an obvious synonym pairs (like “10” with “Ten”) and by adding the redundant category information to the query which contains a single entity, as in ”Tom Cruise” and “Tom Cruise the actor”. The game “*Rephraser*” which we are introducing in this paper allows us to take advantage of the wisdom of crowds in the process of para-query generation. The game implements the “you know it when you see it” notion of semantic equivalence and generates pairs of para-queries. If N users independently propose p as a para-query of a given start query s, we can assume the para-queries are equivalet. The higher N is, the more confidence (consensus) we have in confirming the semantic equivalence of p and s. We call such p as the N*-confirmed* para-query of s.

Due to the small local deployment of our game, 70% of the phrases proposed by participants have $N <= 2$. We believe higher N would be harvested if the game were more broadly played. However, it already showed what we want to demonstrate - even though users easily see semantic equivalence, search engines very rarely do. In our experiments, N is tuned to be 3, which means 4 people typed the same phrase for a start query in the game.

In order to maximize the utility of the game we choose some start queries of the game to compose templates. For example, the query “what's the health benefit of MILK” can be viewed to belong to the template “what's the health benefit of [X]”. The argument slot X could be food, fruit or anything consumable. Any para-query of “what's the health benefit of MILK” would also be a para-query for “what's the health benefit of ORANGES”. One can generate countless numbers of para-queries from templates of different domains with different arguments (types). In this way, one can conduct massive experiments to measure the semantic invariance of search engines. For our experiments we have collected 780,298 top URL results for different para-query pairs.

In the remainder of the paper, we review related work, present three methods for English para-query generation, discuss details of the *Rephraser* game, propose the paraphrasing invariant coefficient and entropy metrics to measure the ability of para-query recognition of four search engines and draw conclusions from experiment results.

2. RELATED WORK

Para-query generation is a sub-class of paraphrase generation. Previous work on automated paraphrasing have thoroughly studied synonym substitution via distributional similarity [14][11]. Synonym substitution is a general method to generate paraphrases. Coyne and Ranbow [5] focused on verb synonyms and built an English paraphrase lexicon “LexPar”. We used numeric synonyms like “10” and “ten” to compose many para-query pairs in [7]. One disadvantage of synonym substitution is that the number of paraphrases is limited to the number of synonyms. This method only causes minimal changes in syntax.

Another approach for paraphrase generation is template induction: identifying templates in large un-annotated corpora. Lin and Pantel [12] parsed text fragments and extracted semantically similar paths in order to derive inference templates. However, they only considered templates with two arguments, for example “X wrote Y” and “X is the author of Y”. Shinyama et al. [16] relied on dependency tree information to extract templates. Their data is articles about the same event from different news resources. In each case, the main idea is to extract templates from input data. Input data is critical and decides the characteristics of templates. Their results showed that most of their templates reflect common syntax changes, such as passive tense and active tense.

Other similar methods use lexical and syntactic information too. Such methods don't work well for our para-query generation problem. Our queries are question-based. They don't contain many useful content verbs or nouns, which makes lexical or syntactic transformation hard. Another direction for paraphrase generation is to borrow ideas from statistic machine translation [15]. Paraphrase generation can be viewed as a monolingual machine translation problem. But this method is not appropriate for para-query generation, because it requires large parallel or comparable corpora to train on and its goal is to find the most likely paraphrase not to find as many paraphrases as possible.

As to para-queries, there doesn't exist enough assistant information to serve as annotated parallel or comparable training data set. Moreover, queries are short question-based sentences which don't contain many words that could be replaced by their synonyms. More importantly, we need a method which can collect as many as possible para-queries to make our experiments objective. Inspired by the concept "game with a purpose" [17], we developed a game "Rephraser" to collect para-queries.

Game with a purpose, the idea of channeling collective human knowledge over networks to solve difficult problems, has had great success. One of the first and most famous realizations of this idea is an online game called ESP [18]. Two players see one image at the same time and try to guess how their partner would describe the image. Only when two players agree on the annotation of the image, will they move on to the next image. The shortcoming of ESP is that players have to agree on as many images as possible in 2.5 minutes. Thus players are motivated to provide short words and only consider the obvious part of the image.

TagATune[9] is a similar game for sound annotation. Players have 3 minutes to come up with agreeing descriptions for as many sound clips as possible. Picture This[2] is another two-player game to agree on the best images for a given query. Players only need to guess one out of 2-5 images which would be chosen by their partners. The problem is that two players have at least a 20%-50% percent chance of choosing the same image for any given query. In each round of the game, the number of images to choose is much smaller than the number of image results for a real query. None of these three games can support more than two players at each round.

The Game designed by Mandel and Ellis [13] can be played by many players simultaneously. Players come up with words to describe a sound clip. If one of their words is used by another player, they will score a point. However, if more than 2 people used the same word, they will get zero points. That means they don't encourage people to agree on the same word because their goal is to collect many different descriptions to a sound clip.

Our game supports many players at the same time. Players are motivated to come up with a para-query that will be agreed on by more players. In image tagging, sound clip tagging and synonym labeling games, players always get assistant information, such as images, sound clips or synonym candidates. Such assistant information reduces the game's difficulty and gives a bias to players. In our game, only a start phrase is given so players will be less likely to be led to the same rephrases by shared assistant information. They must independently generate para-query that accurately represent the meaning of the start query and are also likely to be repeated by other users.

3. PARA-QUERY GENERATION

In this section, we discuss three methods to generate massive para-queries: synonym replacement, augmentation and *Rephraser* game. Currently, we only consider para-queries in English.

3.1 Synonym replacement

Synonyms are words or multiple words that can replace each other in some class of contexts with an insignificant change to the whole text's meaning. Synonym replacement is broadly used in paraphrase generation. The method is to generate a paraphrase p for a source sentence s by substituting some words in s with their synonyms [3]. As we described in [7], we measured the sensitivity of search engines to the use of synonym-based para-queries. There exist absolute and relative synonyms. For instance, these are a group of relative synonyms:

> schedule, plan, design, map out, project, lay on, scheme

And these words are examples of absolute synonyms:

> USA, the United State of America, United States

Absolute synonyms are absolutely semantically equivalent. In order to avoid possible ambiguity, we focused on the measurement of absolute synonyms.

Instead of generic synonym replacement, we worked with simple numeric synonyms such as "ten" and "10" to avoid a lot of human work. Queries related to numbers account for a large percentage of queries. We extracted 1500 queries starting from "top [number]" from AOL query logs. Such fact serves as a hint that using numeric synonyms is a simple and effective method for para-query generation. Although it may be less likely that a user would compose a query as "top one hundred songs" instead of "top 100 songs", we argue that a truly semantic search engine should return the same results to both. Numeric synonyms are one of the easier absolute synonyms to generate, so we used numeric synonym replacement to generate a large set of para-queries.

3.2 Augmentation

Adding extra descriptors to an object in a conversation is usually beneficial in human dialogs. For example, adding a category to the object helps the listener better understand the context of the object. For semantic search engines which are expected to mimic human's understanding of queries, submitting queries like "IBM the company" instead of "IBM", "France the country" instead of "France" or "Tom Hanks the actor" instead of just "Tom Hanks", should not change the set of results. By augmenting redundant category information to entities, we composed 7,198 pairs of para-queries. It's also a straightforward and effective method to generate para-queries. This method is similar to the work on query reformulation [4]. However, as to query reformulation, the goal is to improve the relevance of a retrieval system by adding extra terms. Those extra terms may not preserve the original meaning of the query. Our goal is to achieve the equivalence of the meaning. So we carefully chose entities and categories and attempted to avoid ambiguity and irrelevance. Our results show that the addition of redundant information does not help the understanding of queries for today's search engines.

3.3 Rephraser game

3.3.1 Game Design

The *Rephraser* game which we prototyped at *Ask.com* is intended to produce massive numbers of para-queries by a real-time competition played by two or more players. The objective of playing the game is to guess a hidden phrase which is “semantically equivalent” to the start phrase. When a game starts, the start phrase is given to all participants of the game at the same time. All players enter as many equivalent phrases as possible until the 15 minute game ends. Players receive points by two ways: by getting votes of their rephrases from other players, and by matching the hidden phrase.

A phrase is only valuable if other participants independently submit that same phrase later. This is similar to the ESP game of image labeling where two players have to agree on an image tag in order for the tag to be an adequate description of image. The number of participants repeating the same phrase constitutes the *Votes* for that phrase. The more *Votes* a phrase receives, the more points the owner of the phrase gets. If a participant is the first one who proposed a particular phrase, we call that new phrase a *patent* and the player who owns the *patent* gets extra bonus points. Participants do not see phrases submitted by others. Thus, a large value of *Votes* for a phrase is a strong confidence signal of being a para-query since it reflects the large agreement by participants. A player can also gain points by matching the hidden phrase. We call it hitting the jackpot. Having a jackpot makes the games more interesting to play. At the end of the game, whoever gets the most points wins the game.

3.3.2 A Game Scenario

We used NASCAR racing theme as the design for the game. During a game, the progress of each player is shown by the position of his/her racing car. Figure 1 shows one of many rephrasing games we played. The start query is “How many species of SNAKEs are there”. Figure 1 shows when the player “Jon Goode” receives two points from his phrase “species of snakes” because two other players repeated the same phrase. The Green Progress bar in the Score History section gives hint on how close the guessed phrase is to the hidden phrase. The hidden Phrase of this game was “how many different kinds of snakes exist”. If a player typed the same phrase as the hidden phrase, he/she would receive 20 bonus points. After hitting the jackpot, the player continued to play until the end of 15 minutes. In this game, there were 12 players, 257 patents and 97 of the patents were scored with at least one votes. Some of scored patents from this game are listed in table 2.

EnteredPhrase	Votes
how many different kinds of snakes exist	7
how many different kinds of snakes are there	6
how many kinds of snakes exist	5
how many kinds of snakes are there	5
how many different types of snakes are there	4
how many different snakes are there	4
how many types of snakes are there	3
how many kinds of snakes exist in the world	3
how many different species of snakes are there	3
how many types of snakes exist	3
how many types of snakes are there in the world	3
how many different kinds of snakes	3
how many different kinds of snakes are there in the world	3
How many different species of snakes are there in the world	3
species of snakes	2
snake species	2

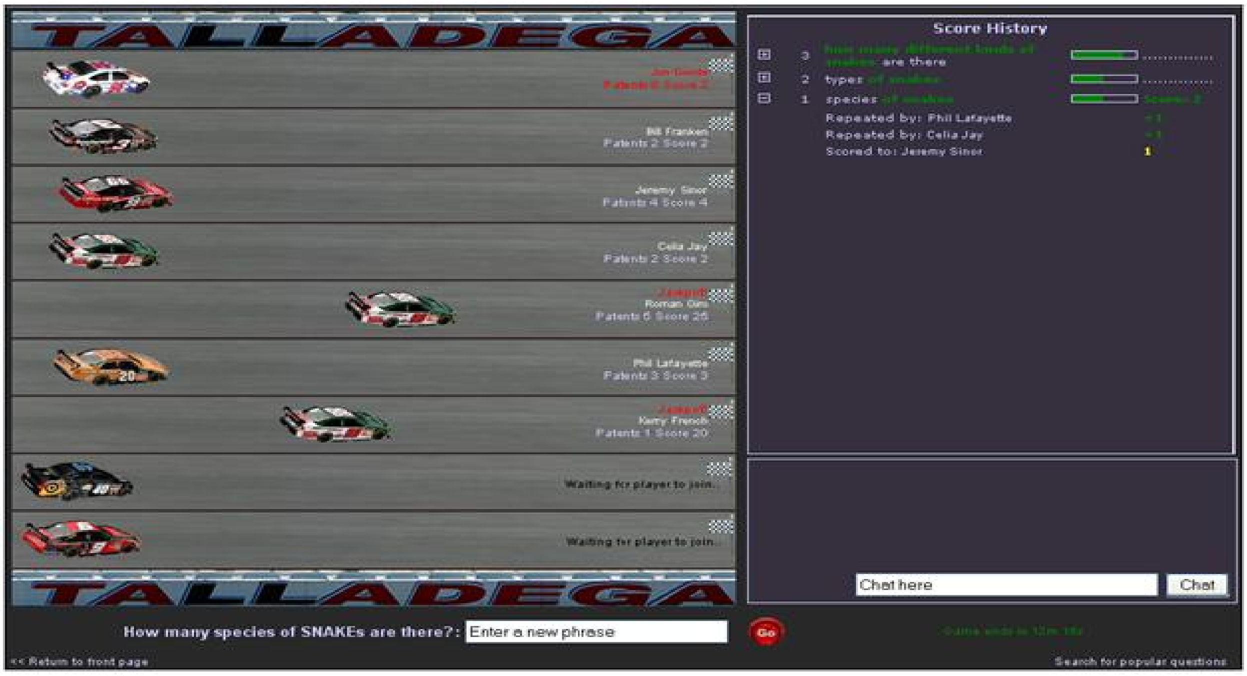
Table 2: Scored Patents from the game

3.3.3 Extracting Para-query Templates

According to previous work on paraphrase generation, the semantic quality of paraphrases are checked by hand [1] [10] [8]. The regular setting is two judges(at most four) and a set of paraphrases. Each judge independently examines each pair of paraphrases and votes “TRUE” or ”FALSE”. Only paraphrases which get “TRUE” vote from all judges are considered as accurate ones. In our game scenario, each participant independently proposing para-query candidates can be seen as voting “TRUE” to those phrases as well. In this sense, each participant serves as both the player and the judge. Thus our game is inherently human validated because the participant will get points only if other people agree on his proposal. Such agreement among participants ($Votes >= 1$) can used to evaluate the quality of our para-query generation. “$Votes = 2$” means two people submitted the same phrase as well. Based on the value of *Votes*, we can filter out low quality para-queries.

Several participants independently proposing one phrase is a strong signal that this phrase is semantic equivalent to the start query. However, if a phrase doesn’t get more than 1 vote, we can’t say for sure that it’s not a para-query because it can’t be seen by other participants. If they could see this phrase, they may vote for it too. Since votes in this game have stronger influence in agreement, each participant won’t be affected by others’ proposals. Such a design makes phrases shared among participants more valuable. Moreover, the number of judges in our situation is larger than 2. The average number of participants in 430 rounds of games is 7. We can’t simply apply either the human acceptability ratio or Kappa to measure the quality of para-queries due to the much more complex setting. Kappa assumes that the sum of the probability to vote each category equals 1, but it is not true in our case. We have no idea how many paraphrases exist for a given start query because of the flexibility of natural language and we can’t claim that para-query candidates proposed by participants in the game cover all possibilities. Therefore the assumption of Kappa doesn’t hold for our case. Using the simple average of human acceptability(the ratio of the number of agreeing judges to the total number of judges) to measure, our phrases with $Votes >= 3$ gain 82% agreement. The result is much better than the baseline for paraphrase generation, which is around 60%-70%.

Based on the results of the human acceptability ratio, $Votes = 3$ can produce plausible para-queries. To balance the value of Votes and the number of participants, we simply use the $Votes = 3$ as the filtering threshold. We are still working on the method to select the minimal acceptable threshold for the number of votes: given M players and K votes, how large should be K in order to guarantee that the phrase with K votes is indeed a para-query with



3 how many different kinds of snakes are there

2 types of snakes

1 species of snakes

Repeated by: Phil Lafayette

Repeated by: Celia Jay

Scored to: Jeremy Sinor 1

How many species of SNAKEs are there?: Enter a new phrase

Figure 1: A snapshot of game

high confidence.

We extracted para-query templates for our experiment in two steps.

- we set a threshold for the value of *Votes* of proposed phrases for every start query. The threshold is tuned to be 3 which means 4 people including the first proposer agreed that this phrase has the same meaning as the start query.
- we manually chose start queries which could form templates easily. For instance, ”Who is the governor of ALASKA now” can be inducted to the template ”who is the governor of X now”. Candidates for the argument slot X could be 50 states in U.S.A.

We selected 14 start queries and their corresponding para-queries to compose templates. Before expanding templates with massive data, we normalized para-queries to collapses together queries which are obviously equivalent(eg: queries differ only in character cases).

Our para-query templates have a large amount of lexical variation. Using the bag-of-words model for para-queries corresponding to start queries, we built a matrix for them and used term frequency as features. Then cosine similarity is applied to the start query and its para-queries. Unsurprisingly, the average similarity score is 0.45. The lowest score is 0.2. The reason is that the para-queries we collected share few common words. For example, the paraphrases “what causes [X]”, “why do I get [X]” and “the reason for [X]” only share 1 word–the slot argument–“[X]”. The low similarity score reveals the lexical variety of our para-queries. Methods in previous work focusing on distributional similarity [8] wouldn’t consider these phrases as paraphrases. In addition, queries are much shorter than sentences in a document and no context information is available to facilitate understanding. Para-queries don’t require 100% grammatically correctness because normally search engines don’t require it. Therefore, previous work on paraphrase generation such as multiple sequence alignment can’t handle these challenges in para-query generation. Our game provides an efficient way to automatically collect para-queries from which it’s easy to generate para-query templates. The main concept of our method is to trust the wisdom of crowds. By filling argument slots, we got 19,518 pairs of (start query, para-query).

4. TESTING SEMANTIC INVARIANCE

4.1 Paraphrasing Invariance Coefficient

Semantic equivalence is a relationship between two queries and is independent of the search engine. Each search engine defines multiple relationships between queries through the search engine return page (SERP). For instance, two queries may share the top result in the SERP, they may share the top K results but not necessarily in the same order, or finally, they may share the top K results, all in the same order.

In this paper, we only consider the top result to simplify the problem. If the search engine S returns the same top URL for para-queries q_1 and q_2, we view the search engine is able to recognize the equivalence of q_1 and q_2. As we said, one can define many other more or less strict relationships between q_1 and q_2 through a search engine S. Here we only consider the same return URL in the top position. The *paraphrasing invariance coefficient*(PIC) is defined to measure how well a search engine recognizes para-queries. It is the probability that top result will stay the same after

Pair number	Google	Bing	Yahoo	Ask
1623	0.49	0.4	0.13	0.47

Table 3: PIC for Numeric Synonyms

Pair number	Google	Bing	Yahoo	Ask
7198	0.360	0.52	0.507	0.510

Table 4: PIC for Augmented Queries

paraphrasing for a given search engine.

Given any two para-queries (q_1, q_2), we define the PIC as following:

$$PIC(S) = Pr(q_1 S q_2 | q_1 \approx q_2)$$

where $q_1 S q_2$ means q_1 and q_2 have same top URL returned by search engine S and $q_1 \approx q_2$ means q_1 and q_2 are para-queries. To calculate such probability, we first compute the PIC of para-queries in a given set $Q = (q_1, q_2 .. q_n)$ by below equation.

$$PIC(Q, S) = \frac{m}{N}$$

where m is the number of para-query pairs which yield same top result by the search engine S; N is the number of all para-query pairs in the set Q. Of course we cannot find the real value of $PIC(Q, S)$ since we do not have a complete list of all para-query pairs. However, we can try to estimate it. Section 3 provides details of three methods to generate a large number of pair-queries. We estimate the PIC value for search engine S by taking the density of many sets of para-query pairs:

$$PIC(S) = \frac{\sum_{i=1}^{K} PIC(Q_i, S)}{K}$$

where Q_i is one set of para-query pairs and K is the number of sets.

As we mentioned in the paper[7], we composed semantically equivalent query pairs by replacing numeric synonyms, for instance "top 10 songs" and "top ten songs". In table 3, we show the $PIC(S)$ result for each search engine testing on synonymous queries. We also consider queries and their augmentation forms, such as "tom cruise" and "tom cruise actor". In table 4, we show the $PIC(S)$ result when using augmented queries as the test set.

In table 5,we present the result of $PIC(S)$ testing on para-queries generated from the *Rephraser* games. Words that can be replaced by other slot arguments are marked in boldface. As we can see in table 5, for the first template "what causes a [headache]", after paraphrasing this query into one of its para-queries, the probability that the search engine will yield the same top result is only 0.11 for Google and 0.08 for Yahoo. Thus, there is a 89%-92% chance that the top result in these search engine result pages will change with simple paraphrasing of a query. This is unacceptably high for a semantically invariant search engine! It shows how sensitive search engines are today to the way the query is formulated. Even a small, obviously semantically equivalent rephrasing will change the top result with high probability.

Templates	Google	Bing	Yahoo	Ask
what causes a **Headache**	0.11	0.10	0.08	0.14
when was **Mark Twain** born	0.52	0.24	0.26	0.55
who wrote **THE BIBLE**	0.28	0.36	0.21	0.18
where can I buy **VIAGRA**	0.13	0.18	0.14	0.13
currency of **RUSSIA**	0.21	0.25	0.18	0.19
how much do **TEACHERS** make a year	0.38	0.26	0.19	0.35
where is **RUSSIA** located	0.15	0.17	0.13	0.10
Is **COPS** on tonight?	0.29	0.35	0.28	0.22
fun things to do in **NYC**	0.09	0.09	0.05	0.37
what are the health benefits of **MILK**	0.25	0.15	0.10	0.22
Daytona 500 pole winner	0.26	0.25	0.16	0.27
What's the oldest CITY in the **U.S.A**	0.26	0.29	0.23	0.22
Who is the governor of **Alaska** now?	0.50	0.53	0.35	0.39

Table 5: Paraphrase Invariance Coefficients

Notice that $PIC(S)$ varies widely for different templates. For example, for the second template "when was [Mark Twain] born", the PIC value for google is 0.52, which means Google will return the same top result for this query more than half the time after paraphrasing. Overall there is about a 66% chance that results for any template will be affected by query paraphrasing.

4.2 Entropy

Entropy is a well-known statistical standard [6] in Information Theory to measure the level of uncertainty. In this experiment, given a set of para-queries, we want to evaluate the degree of uncertainty of the first URL returned as results by a search engine. Let $p(u)$ to be the probability that a URL u is agreed to be top URL for para-queries P by search engine Si. The entropy for search engine Si and para-queries P is:

$$Entropy(P, Si) = -\sum_{1}^{n} p(u) * \log(p(u))$$

For instance, suppose we have 9 para-queries:

$$P = (q_1, q_2, q_3, \ldots, q_9)$$

Their corresponding returned top URLs are:

$$(u_1, u_2, u_1, u_3, u_1, u_4, u_5, u_1, u_3)$$

Thus, the probability of each u_i being the top returned URLs is

$$p(u = u_1) = \frac{4}{9}, p(u = u_2) = \frac{1}{9}$$

$$p(u = u_3) = \frac{2}{9}, p(u = u_4) = \frac{1}{9}, p(u = u_5) = \frac{1}{9}$$

The entropy would be calculated as:

$$H(u) = -\frac{4}{9} * log_2(\frac{4}{9}) - \frac{3}{9} * log_2(\frac{1}{9}) - \frac{2}{9} * log_2(\frac{2}{9}) = 2.06$$

Templates	Google	Bing	Yahoo	Ask	MAX
what causes a **Headache**	2.60	2.53	2.70	2.42	3.17
when was **Mark Twain** born	1.24	2.19	2.16	1.16	3.7
who wrote **THE BIBLE**	1.84	1.53	2.10	2.14	3.17
where can I buy **VIAGRA**	1.95	2.0	2.04	1.88	2.32
currency of **RUSSIA**	1.69	1.67	1.97	1.79	2.32
how much do **TEACHERS** make a year	1.35	1.75	1.94	1.42	2.8
where is **RUSSIA** located	2.58	2.03	2.43	2.40	2.8
Is **COPS** on tonight?	1.55	1.4	1.58	1.83	2.6
fun things to do in **NYC**	2.92	2.94	3.26	1.74	3.7
what are the health benefits of **MILK**	1.95	2.40	2.60	2.06	3.17
Daytona 500 pole winner	1.74	1.82	2.13	1.71	3
What's the oldest CITY in the **U.S.A**	1.73	1.63	1.85	1.93	2.8
Who is the governor of **Alaska** now?	1.14	1.0	1.53	1.49	3.17

Table 6: Entropy

We then expand each group of para-queries by filling argument slots in templates. The entropy is the average value of all groups of para-queries. Table 6 is the result of our entropy measurement.

In table 6, under each search engine's column, every cell represents the entropy of each search engine for each template. The last column "MAX" represents the entropy value if no URL is agreed on by any two para-queries. Entropy represents the "potential for disorder" in a system where smaller entropy means smaller disorder. Semantically equivalent queries are supposed to receive the same answer. The ideal semantic search engine shouldn't have the "disorder" in its top result, therefore the entropy for ideal semantic search engine should be 0. A high entropy value implies today's search engines are quite "disordered" for semantically equivalent queries. The template "Who is the governor of [Alaska] now" has the best entropy value among all templates. We found that for para-queries in this group, most of top returned URLs are government web sites such as "http://gov.state.ak.us". We guess that state names of the U.S.A. could be recognized by most search engines thus they will rank the government sites to the top. An additional observation from our result is that different search engines' behaviors to para-queries are quite similar. The difference on entropies for the same template but different search engines is not significant. However, domain(group) information affects search engines' results. In table 6, for every search engine, entropy fluctuates when the domain information varies (different rows).

5. CONCLUSION AND FUTURE WORK

In this paper, we defined the paraphrasing invariant coefficient as a standard metric to measure para-query recognition of a search engine. In the future, we are trying to find more general and standard measurements to test semantic invariance of search engines. For a search engine to be truly semantic, it is necessary for the PIC to be as close to one as possible, which demonstrates the concept that not matter how the question/query is paraphrased the result(s) should be invariant to such paraphrasing. Future work would be to investigate whether users need such semantic search engines.

We also discussed how to generate monolingual para-queries. Besides synonym replacement and augmentation, we took advantage of "games with a purpose". The idea of "games with a purpose" exploits human interest in online games to address large scale, non-computable problems. Here the *Rephraser* game helps us to collect para-queries. With the help of the game, we generated a great number of para-queries to conduct objective experiments. Our game was locally deployed. In the future, the game would be played by more people and gain more information. Another interesting problem would be to solve the evaluation problem mentioned in the section 3.3.3. It would be valuable if we could find a correct math model to describe our scenario so that we can fairly measure the quality of our game.

6. REFERENCES

[1] R. Barzilary and L. Lee. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *HLT/NAACL*, 2003.

[2] P. N. Bennett, D. Maxwell, and A. Mityagin. Learning Consensus Opinion: Mining Data from a Labeling Game. In *WWW*, Madrid, Spain, 2009.

[3] I. A. Bolshakov and A. Gelbukh. Synonymous Paraphrasing Using WordNet and Internet. In *NLDB*, 2004.

[4] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic Query Expansion Using SMART: TREC 3. In *the Third Text REtrieval Conference (TREC-3)*, 1995.

[5] B. Coyne and O. Rambow. LexPar: A freely available English paraphrase lexicon automatically extracted from FrameNet. In *ICSC*, 2009.

[6] R. M. Gray. *Entropy and Information Theory.* Springer-Verlag, first edition, 1991.

[7] T. Imielinski and A. Signiorini. If you ask nicely, I will answer. In *Proceedings of the 3th IEEE Semantic Computing*, Berkeley, CA, 2009.

[8] D. Kauchak and R. Barzilay. Paraphrasing for Automatic Evaluation. In *Human Language Technology Conference of the North American Chapter of the ACL*, 2006.

[9] E. L. M. Law, L. von Ahn, R. B. Dannenberg, and M. Crawford. TAGATUNE: A game for music and sound Annotation. In *OCG*, Austrian, 2007.

[10] Y. Lepage and E. Denoual. Automatic generation of

paraphrases to be used as translation references in objective evaluation measures of machine translation. In *IWP*, 2005.

[11] D. Lin. Automatic retrieval and clustering of similar words. In *ACL/COLING*, 1998.

[12] D. Lin and P. Pantel. DIRT - Discovery of inference rules from text. In *SIGKDD: 323-328*, 2001.

[13] M. I. Mandel and D. P. W. Ellis. A WEB-BASED GAME FOR COLLECTING MUSIC METADATA. In *OCG*, Austrian, 2007.

[14] F. Pereira, N. Tshby, and L. Lee. Distributional clustering of English Words. In *ACL*, 1993.

[15] C. Quirk, hris Crockett, and W. Dolan. Monolingual Machine Translation for Paraphrase Generation. In *EMNLP*, 2004.

[16] Y. Shinyama, S. Sekine, and K. Sudo. Automatic paraphrase acquisition from News Articles. In *NAACL-HLT*, 2002.

[17] L. von Ahn. Game with a purpose. In *IEEE Computer Magazine*, 2006.

[18] L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. In *ACM CHI*, Vienna, Austria, 2004.