LEARNING BY ANALOGY IN A HYBRID ONTOLOGICAL NETWORK

Ponomarev S. V. (serv@newmail.ru)

Sputnik LLC, Moscow, Russia

This article describes the general principles of question-answering (QA) system, which produces answers to questions by analogy with the answers and the questions at training sets. As a knowledge base the system uses a number ofontologicalinformationofwordsandexpressionsfromopen-accesssources and statistic information, collected by processing large text corpora.

The knowledge base is presented as a hybrid ontological network—an oriented graph, where vertices[1] are the words and expressions and edges are the links between words. In addition, each link between two words or expressions is oriented, typified and weighted. The link type characterizes the information source, from which this link and its type were extracted (for example, synonym from Wiktionary). Link weight is determined by reliable information source. All links, obtained from dictionaries and ontological bases, have the weight equals to one. The links, collected by processing text corpora, have the weight equals to frequency of relevant agreed bigrams (for example, a bigram adjective + noun).

The structure of the hybrid ontological network characterizes by a large number of links between the network vertices. Besides direct links connecting two particular network vertices, there could be used composite links, passes through intermediate vertices, which leads to cardinally increasing of number of possible ways between vertices.

Here's a training algorithm that allows setting in the hybrid ontological network the links between words and items in term of combinations of weighted paths between network vertices.

Key words: ontology, linked data, query answering, semantics

1. The review of ontology systems with Natural Language interface

Systems with Natural Language interface can be divided into two groups—the first, Natural Language dialog with the user oriented (QA systems) and the second—those using Natural Language information sources to detach from the text entities and relations, for mapping into ontological databases.

QA systems, for example, QASIO [2] ontology-based domain-specific NLQA [3] and cross ontology QA on semantic Web [4], use translation of the Natural Language request into the requests for ontologies format. Both SPARQL and ones' own query languages can be used for execution of requests for ontologies.

[1] https://en.wikipedia.org/wiki/Vertex_(graph_theory)

The translation is realised using the formal rules that pose corresponding request template for each possible type of requests, for example:

Who	asking what or which person or people (subject)	PERSON
How far	asking about distance	NUMBER
How many	asking about quantity (countable)	NUMBER
How much	asking about quantity (uncountable)	METRICS

Thus, QA systems described realise the function of increasing of friendliness of ontology access for user, without changing the ontology data itself.

From the other side, the industry's present-day task is to translate inner technical documentation into the machine-processable form and integrate different contractor's documentation into the unified ontology, for unification of the information access.

Data Engineering Methodology of the ISO 19526 [5] standard regulates the forms of integration and processing of technical information from different sources.

The works were done of automatic parsing of Russian documents with excretion of entities and the relations between them with the use of ABBYY Compreno [6] technology. The parser accepts at the input Natural Language technical text and brings its mapping to existing ontologies.

The impossibility to eliminate all the ambiguities inherent to Natural Language, is the factor that limits possibilities of this approach. Consequently, mapping of technical text into ontologies cannot be univocal, it should represent statistically probabilistic structure.

Accordingly, methods processing such ontological data should consider the ambiguity and, perhaps, the in coordination of mapping that was built.

This paper shows the approaches that allow to substitute hand production of Natural Language request analysis rules by methods of teaching by examples. And also—approaches of solving the ontology data ambiguities by means of combination of different sources ontological data and use of relations that have probabilistic nature.

2. The Hybrid Ontological Network

Open-access ontologies in Russian do not involve the processing of links with probabilistic nature, because in these ontologies indication of the triplet weight doesn’t provide. Accordingly, these ontologies can’t be expanded by the links, accumulated in statistical text corpora processing, and merging them with other information sources is difficult of discrepancies between different information sources. Adding to the triplet value of its confidence (weight) allows solving these problems.

We call an ontology hybrid if:

1. It is composed of several independent sources;
2. It contains triplets (links), accumulated in statistical text corpora processing;
3. Each triplet characterized by type and weight.

The main properties of this ontology are redundancy and high relatedness. Redundancy arises from the duplication of most ontological links in various used sources, and high relatedness arises in inclusion of links obtained by statistical text processing.

If we represent this ontology as a network with vertices-concepts and edges-links, then the hybrid ontological network characterizes by a large number of possible paths between network vertices, including through some intermediate vertices. The use of statistical data ensures that not even listed concepts in used information sources, for example, rare words or names connect with other network vertices by sufficient number of links.

The total number of vertices in the network 1,355,135 and summary of link types of the hybrid ontological network are shown in Table 1.

Table 1. Structure of the hybrid ontological network

N	Link	Number of links	Link type	Source
3	Idiomatic expressions	9,334	Onto-logical	Wiktionary [7]
4	Epithets	49,929		
5	Antonyms	24,900		
6	Synonyms	739,053		
7	Hypernyms	29,545		
8	Hyponyms	30,871		
9	Higher category	12,332		
0	Set phrases	16,068		
13	Related words	407,895		
14	Holonymy	475		
15	Meronymy	667		
10	Categories	226,800		
24	Examples of use	16,463		
2	Defining words	4,672,480	Statis-tical	Defining words are words from articles of dictionaries for which the ratio of the frequency of words in the article to the frequency of words in the whole corpus ratio is as large as possible.
12	Homonym relations	17,092	Statis-tical	Homonym relations are set between vertices by comparing all possible word grammatical forms.
11	Words are included in one phrase	231,416,665	Statis-tical	Uncoordinated N-grams obtained by parsing news corpus
30	Word is adjacent to the left	22,551,832		

N	Link	Number of links	Link type	Source
17	N-gram noun + noun “house of cards”	28,722,993	Statistical	Collecting statistics using the SDK Grammatical dictionary [8]
18	N-gram adverb + verb “work hard”	2,148,646		
19	N-gram adverb + adjective “very good”	1,722,124		
20	N-gram preposition + noun “at the table”	623,370		
21	N-gram verb + managed object “see a mouse”	4,234,149		
22	N-gram adjective + noun “spiral galaxy”	10,249,513		
23	N-gram noun + verb	7,518,027		
25	The phrase is composed of	951,895	Internal	Network vertices of several words (collocations and phrases) have links with the words of which they consist.
26	The first word of the phrase	310,817		
27	The second word of the phrase	310,817		
28	Number of instance of a word to the phrase	934,023		
29	Number of instance of a collocation to the phrase	228,386		

Described structure of the hybrid ontological network allows to set the relationship between network vertices of various types, for example—“synonym” or “attribute value”. At the same time, the links are characterized by computable confidence level in the range [0, 1]. In other words, established relationship is essentially a classifier that estimates whether there is a relationship between the real-world entities on the basis of available information on the network.

3. Automatic relation building

Automatic relation building is based on training sets. Relations, collected from training results, allow QA system to form the response in a manner similar to the method of forming the answer to the question in a learning sample. The right relation between question and answer in the learning sample is unknown, as there is only pair “question and answer” available for training without comments of what conclusions have led person to this particular answer. Thus, a well-formed relation should outwardly repeat structure of human conclusions.

Relations are set paths between network vertices, encoded as an index sequence of link type. For example, the link "TOMATO >> COLOR" can be coded as follows: "TOMATO >> coherent n-grams "noun + adj."(–22) >> RED >> hypernym (7) >> COLOR". In case of arbitrary start and target vertices: «START >> coherent n-grams "noun + adj." (–22) >> RESULT >> hypernym (7) >> FINISH». At that, this network path is not the only one, and path variety between the start and target vertices may be obtained by passing through another link types. If we use composite paths passing through intermediate vertices, then the total number of possible paths between two network vertices increases like an avalanche.

Each of the paths may be weighted by appropriate coefficient. Thus, the path leading to the correct result may have an increased coefficient, as paths that do not lead to a correct result—have a reduced coefficient or being deleted.

Let's look at example. Given a triple values "TOMATO", "RED", "COLOR".

Table 2. The link structure between the vertices: "TOMATO", "RED", "COLOR"

	Number of links	**Links lead to "RED"**
TOMATO	23	1. Related n-grams "adjective + noun", back link (–22).
COLOR	30	1. Hyponyms (8); 2. Related n-grams "adjective + noun", back link (–22); 3. Phrase consists of, back link (–25).

Link № 8 "hyponyms" obtained by Wiktionary parsing, which explicitly set the connection "Red is a hyponym of the word Color". Link № 22 is derived by statistical text processing, as agreed bigram "red" and "color". As we move from the word "color" to the word "red", the back link is used (from the color to red). Link № 25 shows that the word "color" and "red" are present together in one of the network vertices. Vertice "Red color" received by parsing dictionaries.

Thus, if we follow from the vertice "TOMATO" to link № –22 and from the vertice "COLOR" to links № 8, № –22 and № –25, then we'll get the following link picture:

Table 3. Building a path between the vertices: "TOMATO", "RED", "COLOR"

	Link type	**Number of links**	**Top 10 of the vertices**
TOMATO	–22	382	**red**, best, fresh, ripe, rotten, sliced, marinated, rotten, green, salty ...
COLOR	8	7	blue, purple, sea color, orange, **red**, brown, green
COLOR	–22	4,032	whole, **red**, white, black, yellow, green, blue, gray, such a, own ...
COLOR	–25	265	versicolour, zinnwaldite, fanal, Black Sea, surah, old gold, cream, blue dust, dark tangerine, light-color ...

To build a path in the hybrid network, you should specify a set of pairs "link type—link weight”:

START >> (–22, 1) >> RESULT;

FINISH >> (8, 1/3) >> RESULT;

FINISH >> (–22, 1/3) >> RESULT;

FINISH >> (–25, 1/3) >> RESULT.

Apply this path to different sets of arguments.

Table 4. Applying constructed path in the network

START	FINISH	RESULT
TOMATO	COLOR	red—0.3744 green—0.2380 most—0.1914 blue—0.1709 fresh—0.1691
CURRANT	COLOR	black—1.1153 red—1.0118 green—0.1941 blue—0.1670 brown—0.1555
CAR	COLOR	red—0.1915 own—0.1789 green—0.1719 blue—0.1683 brown—0.1556
SEA	COLOR	black—0.2317 red—0.2293 blue—0.2233 green—0.1836 mediterranean—0.1570
SEA	SIZE	length—0.2732 high—0.2732 width—0.2500 depth—0.2500 black—0.2026

As Table 4 shows the generated path gives satisfactory results for requests related to the color of the object, but does not apply to other types of requests such as the request of size. For the path formed by the only learning triple “ TOMATO >> RED >> COLOR” it is natural. Try to expand the rule to train it also on the triple “SEA >> LARGE >> SIZE”. In general, it could not pick up such weights for pairs “link type—link weight” without passing through intermediate vertices that outgoing rule satisfactorily completes work on requests associated with both requests: the color and the size.

Let’s consider the building of paths, passing through one intermediate vertice.

Table 5. The link structure through one intermediate vertice

	All link types	**Links lead to "RED"**
TOMATO	606	60
COLOR	1,018	177
	All link types	**Links lead to "LARGE"**
SEA	1,016	107
SIZE	807	108

Further increase the number of intermediate vertices leads to an avalanche-like increase of available links. A lot of links increase the chances of learning algorithm to generate an effective way to respond to a wide class of requests.

Consider the algorithm of the path construction on the hybrid network:

1. Given: one or more training triples "START >> RESULT >> FINISH";
2. Build links such as "START >> RESULT" and "FINISH >> RESULT";
3. Select such weight rates of links, that the desired value "RESULT" was maximum;
4. Carry out a test run: in a path specific values "START" and "FINISH" are substituted of training triples and verify that the maximum value of "RESULT" is the value of teaching triple. If the condition is satisfied, then the path is ready and we exit from the algorithm.
5. Build relations "START >> RESULT" and "FINISH >> RESULT" through one additional vertice;
6. Repeat from step 3.

4. QA system

QA system is trained on pairs "question—answer" given in Russian. Firstly we produce syntactic analysis of question and build syntactic tree. Each type of syntactic trees corresponds one rule at the rule base in QA system. If there is no rule found under questions with the same syntactic tree, then the rule is formed. If the corresponding rule is found, then firstly response is generated. After, the answer is compared with the correct answer, and if they do not match, the rule is extended by another pair of "question-answer" and it's being relearned.

The rule contains the information, which words in question we use as arguments "START" and "FINISH", and information which word in the correct answer use as "RESULT". The bespoke correct answer is broken into words, and for each word is determined by its type: 1) the function words (pronouns, verbs, punctuation), 2) the transfer word (present as in the question and in the answer) 3) the computable word (there isn't in the question, but it can be derived from the question words by building relationships). Thus the system leaves the function words in its place for generation the answer, replaces the transfer words on the relevant words from the question, calculates the computable words and then aligns with the grammatical phrase attributes (gender, number, case). In case of disambiguate, what words to use as arguments to "START" and "FINISH", calculations are carried out for all the variants, and then the system select the path with the least number of links.

Let's look at the example of this approach. We form the rule basis; each rule encodes the output method of answer to a question by analogy with examples from the training set. When you start the system rule base is empty. Each question of the system corresponds the correct answer. The system tries to generate their own answer according to rules base, in case of failure—it remembers a new "question—answer" pair. At the same time, memorized pairs of "question—answer" create a new rule of inference or specify an existing one.

Step 1: The rule base is empty, so random response generated. A pair of "question—answer" memorized.

(1) *Question: Какой глубины лужа?* (What depth is the puddle?)
Correct Answer: Лужа—мелкая. (The puddle is small)
Generated Answer: Глубина. (Depth)
New Rule Added.

Step 2. In the rule base there's the only rule obtained in step 1, and the system tries to apply this rule to the question. Attempt fails and the rule is corrected.

(2) *Question: Какой глубины море?* (What depth is the sea?)
Correct Answer: Море—глубокое. (The sea is deep)
Generated Answer: Море—мелкое. (The sea is small)
Adding 1 New Path.

Step 3. In the rule base there's still the only rule, but it's taught at two examples. The system makes a successful attempt to apply this rule to the question. Thus, in this case two training examples are enough to obtain practically valuable rule.

(3) *Question: Какой глубины океан?* (What depth is the ocean?)
Correct Answer: Океан—глубокий. (The ocean is deep)
Generated Answer: Океан—глубокий. (The ocean is deep)
Correct Answer Found.

Step 4. The syntactic structure of pair "question—answer" is changed, so the use of the existing rule does not give the correct result. Another rule is generated.

(4) *Question: Какой глубины лужа?* (What depth is the puddle?)
Correct Answer: Лужа маленькой глубины. (The puddle is small depth)
Generated Answer: Лужа—мелкая. (The puddle is small)
Generated Answer: Глубина. (Depth)
New Rule Added.

Step 5. Attempt to apply rule № 2, obtained in step 4, gives the correct result within meaning, but not coinciding exactly with the correct answer. Rule № 2 is corrected.

(5) *Question: Какой глубины море?* (What depth is the sea?)
Correct Answer: Море большой глубины. (The sea is deep depth)
Generated Answer: Море огромной глубины. (The sea is vast depth)
Adding 1 New Path.

Step 6. The syntactic structure of pair "question—answer" corresponds more with the rule № 2, than with the rule № 1. The attempt to apply rule № 2 to determine the color instead of the size gives the expected result.

(6) *Question: Какого цвета огурец?* (What color is the cucumber?)
Correct Answer: Огурец зеленого цвета. (The cucumber is green color)
Generated Answer: Огурец зеленого цвета. (The cucumber is green color)
Correct Answer Found.

Similar way the appliance of rule № 2 gives the correct answers to the questions "What color is a tomato?" and "What size is a seed?". The structure of rule № 2 rules is given in Table 6. Total number of paths in rule № 2 is 54 left and 123 right, the most important paths are included to Table 6**.**

Table 6. The structure of rule №2

START	Weight of path	Path	RESULT	Weight of path	Path	FINISH
color depth size	0.2352160	26 0 7	green red big small	0.1685550	3 −7 27 9	cucumber tomato seed sea puddle
	0.1176080	25 0 7		0.1348440	12 −9 24 −16	
	0.1176080	28 0 7		0.1348440	−15 −9 24 −16	
	0.0996208	7 −16		0.0374567	3 −3 26 24	
	0.0958917	3		0.0345754	5 −27 −32 24	
	0.0740494	3 27 0 −8		0.0345754	−8 −27 −32 24	
	0.0282505	−25 27 0 −8		0.0313329	−25 15 7	
	0.0270013	6 0 −10 −23		0.0280925	2 −10 −27 5	
	0.0270013	−9 0 −10 −23		0.0232490	4 −7 27 9	
	0.0270013	−28 0 −10 −23		0.0210694	3 27 −29 12	

The data represented in Table 6 is interpreted as follows: for getting the word "green" from the word "color" we need to build the path "COLOR >> the first word of the phrase (26) >> Set phrases(0) >> hypernyms (7) >> GREEN". Or we can use shorter path "COLOR >> idiomatic expressions (3) >> GREEN". Not all the paths formed the rule № 2 can be built for each pair of arguments "color + cucumber", "depth + puddle" and etc., but the excess amount of paths guarantees to find a sufficient number of paths to separate the correct result. Negative indexes mean back links, so link № 7 is a link from hypernym to hyponym. This link is different from link № 8, because the used data source (Wiktionary) is not complete, and an essential part of back links is not filled.

Let's take a detailed look at the first three paths of links "color—green," "depth—big" and "size—small". As seen, the first link type 25, 26 and 28 is an internal link type between the phrases and their components, words (see Table 1). The basic phrases related to the words "color", "depth" and "size" are listed in Table 7.

Table 7. Some links of the hybrid network

	Color	Depth	Size

The phrase is composed of (25)	white;
	painting;
	flowering;
	blue;
	number;
	yellow;
	protective;
	green;
		container;
		seriousness;
		abyss;
		solidity;
		significance;
		thoroughness;
		serious;
		depth
			height;
			growth;
			coverage;
			border;
			length;
			volume;
			scale;
			measure;

The first word of the phrase (26)	hair color;
	skin color;
	color of languages;
	aquamarine- colored;
	turquoise-colored;
		depth on languages;
		depths of the earth;
		depth of hold;
		depth of inhale;
		nesting depth;
			size on languages;
			yield;
			size of the female pelvic organs in the sagittal section;

Number of instance of a word to the phrase (28)	verbs discoloration;
	verbs color development;
	blue color;
	yellow color;
	yellow color;
		languages;
		deep;
		deeply;
		deep;
		in ancient days;
			measure;
			size adverbs;
			size on languages;
			measure by language;
			enormous size;

For the word "color" links 25 and 28 give the required "green", but the 26th link does not lead to an acceptable result. On the other hand, for the words "depth" and "size" can be seen accordance only with the 28th link type: "at a depth," "large size", and the 25th and 26th links do not lead to direct result. It demonstrates the network redundancy and the rules formed as a set of paths in the network. That means fixity rules in their application to the arguments that have not all affixed link types. On the other hand, the rules in the paths passing through the 26th link means that even the naked eye cannot see sense, a positive effect on the productivity of the final rule turns out.

It is important that the ontological data sources, which the ontological network formed, do not contain links such as “sea—deep” and “cucumber—green.” These links obtained by statistical text processing methods.

Technology demonstrator of deduction by analogy is available at http://servponomarev.livejournal.com/6059.html

5. Quality control of QA system

Rule № 2 (Table 6) as a result of learning in two examples “What depth is the puddle?” and “What depth is the sea?” used to assess the response quality to a set of questions oriented at getting the typical response of an attribute object value. Rule № 2, trained only on “depth”, is used to demonstrate the possibility of generalizing to other types of attributes.

Table 8. The answers to some questions according to rule №2

What taste is the lemon?	The lemon is tart.
What taste is the watermelon?	The watermelon is sweet.
What taste is the herring?	The herring is pungent.
What taste is the onion?	The onion is strong.
What weight is the grain?	The grain is small.
What weight is the cobble?	The cobble is small.
What weight is the bar-bell?	The bar-bell is small.
What color is the cucumber?	The cucumber is green.
What color is the strawberry?	The strawberry is bright.
What color is the lemon?	The lemon is bright.
What color is grime?	Grime is deep.

As seen from Table 8, rule №2, trained by attribute “depth” satisfactorily fulfils also the attribute “taste”, and in some cases the attribute “color”. However, to obtain high-quality results, we should set rules individually for each of the attribute types. For example, the rule formed by the pair question-answer “What color is snow? Snow is white.” shows the following results in mode of relearning according to correct answers.

Table 9. The answers to some questions according to rule “color”

What color is the cucumber?	The cucumber is green.
What color is grime?	Grime is black.
What color is the cloud?	The cloud is black.
What color is the cloud?	The cloud is gray.
What color is the sky?	The sky is grey.
What color is the grass?	The grass is green.
What color is the tomato?	The tomato is green.
What color is the lemon?	The lemon is green.

Performance of large-scale testing hindered by the lack of context, which allows to select the one concrete correct value from the list of valid values. So the answer that the lemon is green is allowable, although more common answer is “The lemon is yellow.” In future versions of QA system we will plan introduction context recording.

6. Quality control of automatic relation building

The method of automatic relation building described in paragraph 2 used in “The First International Workshop on Russian Semantic Similarity Evaluation” [1], where in the category “Evaluation based on Semantic Relation Classification” was obtained accuracy in 0.9209 on criterion Area under Curve (AUC), which ensured 3rd place in the competition.

7. Follow-up research

The research efforts in the direction of automatic selection of the syntactic form of answer to question using only the statistics dialogs without learning by example. We plan to create QA system that generates answers to questions, taking into account the context in its natural form, like a dialogue between two people.

References

1. *Panchenko A., Loukachevitch N. V., Ustalov D., Paperno D., Meyer C. M., Konstantinova N.* (2015) “RUSSE: The First Workshop on Russian Semantic Similarity”. In Proceeding of the Dialogue 2015 conference. Moscow, Russia
2. *Moussa, A. M., Abdel-Kader, R. F.* “QUASIO: A Question Answering System for YAGO Ontology”. International Journal of Database Theory and Application 4(2), 99–112 (2011)
3. *Athira P. M., Sreeja M. and P. C. Reghuraj* “Architecture of an Ontology-Based Domain Specific Natural Language Question Answering System”. International Journal of Web & Semantic Technology (IJWesT) Vol. 4, No. 4, October 2013
4. *Lopez, V., Uren, V. S., Sabou, M., Motta, E.* “Cross Ontology Query Answering on the semantic Web: an Initial Evaluation.” In: Gil, Y., Noy, N. F. (eds.) K-CAP, pp. 17–24 ACM (2009)
5. ISO 15926 Reference Data Engineering Methodology, http://techinvestlab.ru/files/RefDataEngenEnglish/RefDataEngen_ver_3_English.doc
6. The reference data extraction from technical texts in natural languages, http://www.slideshare.net/vvagr/reference-dataextraction
7. Wiktionary, https://ru.wiktionary.org/
8. Russian and English Morphology for Windows and Linux, http://solarix.ru/grammatical-dictionary-api-en.shtml