System mining inference rules from natural language texts

Peeter Lorents
Cooperative Cyber Defence Centre of Excellence

and

Erika Matsak
Institute of Informatics, Tallinn University, Estonia

Abstract. We look at mining inference rules from natural language texts. It is possible to use the DST dialogue system to search for logic formulas and inference rules from natural language texts. In order to explain the extent and reliability of this approach, we use the RS-meta-procedure that was created for system mining.

Keywords. Intelligent systems and logical deduction. Natural language text transformation. Dialogue system DST. Relation and similarity. RS-meta-procedure.

Introduction

The ability to use logical deduction is important for various intelligent systems. Specifically, the ability to apply logic inference rules to get correct arguments. This is necessary in decision support systems, various expert systems etc. Sometimes it is not useful to apply existing systems of inference rules. Non-traditional rules from experts may give much better results in some cases, especially if the expert is using the "reasoning technology" is based on experience from the specific field. However, these experts may not be able to identify and formally describe their deduction mechanisms.

In such cases we could use (dialogue) systems that take the expert's:

1. descriptions (situations, developments) in natural language and transform them into logical formulas and
2. reasoning in natural language and transform it into formal proofs.

This, however, raises the problem of extracting from the natural language texts the formulas needed for formalizing descriptions, as well as the inference steps and inference rules necessary for constructing proofs.

This work aims to describe and compare two approaches: the dialogue system developed by E. Matsak (Matsak 2007, 2008) and P. Lorents' formalization procedure (see Lorents 1993, 2002).

1. Mining formulas from natural language texts with the DST dialogue system

The DST dialogue system is based on the text transformation procedure described by P. Lorents (2000), which consists of the steps described below. The choice of steps, the order of using them and the number of repetitions for any step is not determined beforehand. However, there is the important requirement that *the source text and the resulting text must have the same meaning in each step.* The steps are:

1. Adding something to an existing text
2. Erasing something from an existing text
3. Replacing part of an existing text
4. Reordering some parts of an existing text
5. Identifying the logical roles of the parts in an existing text (for example, to represent an individual symbol, to represent some logic operation or quantifier, modality etc.)
6. Replacing parts that carry a logical role in an existing text with the corresponding logic symbols
7. Arranging logic symbols according to the conventions used for forming formulas.

Using the steps described above helps identify logic constructs that are present in natural language texts.

Example. Let us transform the text "there is nothing terribly sophisticated in here".

There is nothing dreadfully sophisticated in here→(2) *There is nothing sophisticated here*→(3) *There is nothing sophisticated in this text*→(4) *This text has no part that is sophisticated*→(3) *There is no text, which is part of this text and which is a sophisticated text*→(5,6,7) ¬(∃t)[Sect(t,T)&Soph(t)].

In order to mine logic constructs at the formula level, E. Matsak developed algorithms that are described (both structure and realization) in the following publications Matsak 2005, 2006.

2. Mining single or double premise inference steps from natural language texts with the DST dialogue system

Mining inference steps (with DST dialogue system) involves the following steps:

1. Identifying the parts of text that represent an individual or a predicate.
2. Identifying synonyms in the role of an individual.
3. Identifying synonyms and antonyms in the role of a predicate.
4. Replacing the identified antonyms with the negation of corresponding predicates.
5. Characterizing the number of times individuals, predicates and negations occurred with the assessments *occurred once* or *occurred multiple times.*
6. Individuals that occurred once are represented with symbols q1, q2, q3, … .
7. Predicates that occurred once are represented with symbols P1, P2, P3, … .
8. Individuals that occurred multiple times independently or as synonyms are represented with x1, x2, x3, … . All occurrences of an individual and all its synonyms must be assigned the same symbol, for example x8.
9. Predicates that occurred multiple times independently or as synonyms are represented with A1, A2, A3, … . All occurrences of an individual and all its synonyms (including negations of it) must be assigned the same symbol, for example A8.
10. All formulas in the text are transformed so the individuals and predicates are represented with the symbols (and according to the rules) described above.
11. Of all the formulas created in step 10, the ones that contain at least one such atomic formula, which consists of only repeated symbols are selected (for example, A3(x1,x3), but not A3(x1,q3) or P3(x1,x2)).
12. Forming all possible ordered pairs, where the first and second position is filled with the formulas selected in step 11, taking care to not use the same formula for each (this is necessary to avoid trivial inference steps, where some argument would be inferred from itself). However, it is required that the two formulas in the ordered pair share at least one common formula part.
13. Correct single premise inference steps are those ordered pairs from step 12 that meet the correctness criteria (in other words, the ordered pair ⟨F,G⟩ of formulas F and G is an inference step if in case of any interpretation φ, if φF=1, then it follows that φG=1).
14. Forming all possible ordered triples, where the first, second and third position is filled with the formulas selected in step 11 that are not the same (this is required to avoid trivial inference steps, where some argument with the same argument would again result in the same argument). It is required that *at least one of the formulas shares a common formula part with the other two in the triple* (in other words, if we have a triple ⟨X,Y,Z⟩ from formulas X, Y and Z, then at least one of the following requirements must be met: (1) the formulas X and Y have at least one commonly shared formula part and formulas X and Z have at least one commonly shared formula part, or (2) the formulas Y and X, as well as formulas Y and Z have at least one commonly shared formula part, or (3) the formulas Z and X, as well as the formulas Z and Y have at least one commonly shared formula part).
15. Correct double premise inference steps are those ordered triples from step 14 that fulfill the correctness criteria (in other words, a triple ⟨F,G,H⟩ from formulas F, G an H is an inference step, if for whatever interpretation φ, if we get φF=1 ja φG=1, then it follows that φH=1).

(Matsak, 2010)

Example. Let us assume that we have found the following arguments while working on some text: *If the temperature of water is less than zero degrees Celsius, then the water is frozen. However, water is not frozen. Therefore, it is not correct that water temperature is less than zero degrees Celsius.* From these arguments we get to the following triple:

$\langle T(w)<0 \supset F(w), \neg F(w), \neg T(w)\rangle$

It is easy to see that the formula $T(w)<0 \supset F(w)$ has a shared part with the formulas $\neg F(w)$ and $\neg T(w)$. It is also easy to check that they meet the correctness criteria.

3. Mining inference rules from natural language texts with the SR-meta-procedure

(I) Predicates and meta-predicates (Lorents 1993, 2002)

Let as view a non-empty set H. In set theory, the unary, binary and whatever relations or predicates with m+1 connections are defined as follows (see Kuratowski, Mostowski 1967, Chapter II, §6):

P is a m+1 connected relation or a m+1 connected predicate on set H, if P is the m+1 Cartesian order $H^{(m+1)}$ subset of H. In other words,

- in case of unary (m+1=1) predicate $P \subseteq H$ and
- binary, ternary, and in general any m+1 (where m+1>1) connected relation or predicate $P \subseteq H \times \ldots \times H \times H$, where $H \times \ldots \times H \times H$ is a Cartesian multiplication with m+1 components or the set of all such m+1 connected tuples

$\langle h_1,\ldots, h_m, h_{m+1}\rangle$, where $h_1 \in H$, … , $h_m \in H$, $h_{m+1} \in H$.

Definition 1. Set A is the *totality area* of binary relation S, if all elements in set A are related to each other with relation S, or shorter $(\forall xy \in A)[S(x,y)]$

Definition 2. A set M is the *maximum totality area* of binary relation S, if all elements in set M are related to each other with relation S and at the same time no element, which is not part of set M, can have the relation S with all elements in set M. Or shorter, $(\forall xy \in M)[S(x,y)]\&(\forall \alpha \notin M)(\exists \beta \in M)[\neg S(\alpha,\beta)]$

Next we view the set of tuples that can be formed from the elements of set H, $Cart(H) = H^{(1)} \cup H^{(2)} \cup H^{(3)} \cup H^{(4)} \cup \ldots$, (where $H^{(1)}=H$, $H^{(2)}=H\times H$, $H^{(3)}=H\times H\times H$, …).

Definition 3. The predicate Q is called k+1-connected meta-predicate on set H, if Q is a k+1-connected predicate on set Cart(H).

Definition 4. A single connection meta-predicate Rel_H is called the relation meta-predicate on set H.

Note. The meta-predicate Rel expresses the "intuition", "recognition", "vision" etc., where:

- if $x \in H$ and $Rel_H(x)$, then the element x in set H *has some property*
- if $x=\langle h_1,\ldots, h_m, h_{m+1}\rangle$ and $Rel_H(x)$, then the elements $h_1,\ldots, h_m, h_{m+1}$ of set H are *somehow related to each other*.

Definition 5. A two-connected meta-predicate Sim_H is called similarity meta-predicate on set H, if the following requirements are met:

- $Sim_H \subseteq Cart(H)\times Cart(H)$
- Sim_H is *localized by exponent* or, if $Sim_H(x,y)$, then $(\exists k \in N)[\, x \in H^{(k)} \,\&\, y \in H^{(k)}\,]$
- Sim_H is reflexive
- Sim_H is symmetric.

Note 1. Sim_H *does not have to be* transitive. In other words the *transitivity of Sim_H is not required*! Note that if some relation S is transitive then $(\forall xyz)[(S(x,y)\&S(y,z))\supset(S(x,z)]$. Saying "does not have to be transitive", however, does not mean that it *cannot* be.

Note 2. The meta-predicate Sim_H expresses the "intuition", "recognition", "vision" etc., that if for some tuples x and y $Sim_H(x,y)$, then these tuples from the elements of set H are somehow similar to each other.

(II) Mining relations with the SR-metaprocedure (Lorents 1993, 2002)

Let there be some set H and meta-predicates Rel_H ja Sim_H. We presume that $H\neq\varnothing$, $R_H\neq\varnothing$, $S_H\neq\varnothing$.

Let us view the set $Cart(H) = H^{(1)} \cup H^{(2)} \cup H^{(3)} \cup H^{(4)} \cup \ldots$. Since according to Zermelo's theorem (see Kuratowski, Mostowski 1967, chapter VII, §8; Potter 2004, chapter 14, 14.4.3) it is possible to order any set, then we can also order the set Cart(H) and index its elements with suitable ordinal numbers, which are smaller than some α. Let us agree that the symbol h_β is used to represent the element in Cart(H) which is indexed as β. The procedure in question consists of steps based on the indexes so that each for index $\beta<\alpha$ there is a step: *step β*, which, in turn, is divided into sub-steps where each index $\delta<\alpha$ corresponds with a sub-step: *sub-step $\beta\delta$.*

Note. Below, we use the phrases "begin new step" and "begin new sub-step" for ease of understanding. By those phrases we mean the following:

- Each step is indexed with its "own ordinal" β, where $\beta<\alpha$
- Beginning a new step means that we find all such ordinals ρ, where it is known that we have already finished *all steps*, where the index σ meets the criteria $\sigma<\rho$
- Beginning a new step means beginning a step, where the index is ρ.

There is an analogous situation with beginning new sub-steps:

- Each sub-step is indexed by two ordinals $\beta\gamma$, where $\beta<\alpha$, $\gamma<\alpha$ and β is the index of the corresponding step
- Beginning a new sub-step presumes that we find such an ordinal ρ, where it is known that we have finished *all such sub-steps* of step β, where in the index $\beta\sigma$, σ meets the criteria $\sigma<\rho$
- Beginning a new sub-step means in such a case beginning a sub-step of step β, where the index is $\beta\rho$.

Step 0: Forming predicate P_0.

- We take the element $h_0 \in Cart(H)$ and such a natural number m_0, where $h_0 \in H^{(m_0)}$.
- Begin sub-step 00.

Step 0: Forming predicate P_0. Sub-step 00.

- Check the requirement $Rel_H(h_0)$.
- If the requirement is *filled*, then step 0 "Forming predicate P_0" continues and we take $h_{00} = h_0$, $P_{00} = \{h_{00}\}$. Then we begin the next sub-step of step 0.

Comment. Since $H \neq \varnothing$, $Rel_H \neq \varnothing$, $Sim_H \neq \varnothing$ and meta-predicate Sim_H are reflexive, then we can say that the set $\{h_{00}\}$ is the totality area of Sim_H (because $Sim_H(h_{00},h_{00})$ for each element of set $\{h_{00}\}$). Since $h_{00} \in Rel_H$, we can say that the set P_{00} is a intersection of the totality area of the meta-predicate Sim_H and the meta-predicate Rel_H.

- If the requirement is *not* filled, then the step 0 "Forming predicate P_0" ends with the result $P_0 = \varnothing$ and we start a new step.

Step 0: Forming predicate P_0. Sub-step 0δ.

Let us assume that in case of each $\varepsilon<\delta$ there is a set $P_{0\varepsilon}$ formed and the set $P_{0\varepsilon}$ is the intersection of the totality area of the meta-predicate Sim_H and the meta-predicate Rel_H. Let us view the set $H^{(m_0)} - \bigcup_{\varepsilon<\delta} P_{0\varepsilon}$ and check the requirement

- $$(\exists h \in H^{(m_0)} - \bigcup_{\varepsilon<\delta} P_{0\varepsilon})[\mathrm{Rel}_H(h) \,\&\, (\forall h' \in \bigcup_{\varepsilon<\delta} P_{0\varepsilon})[\mathrm{Sim}_H(h,h')]]$$

- If the requirement is met, then the step 0 "Forming predicate P_0" continues and we take

$$h_{0\delta} = h$$

$$P_{0\delta} = [\bigcup_{\varepsilon<\delta} P_{0\varepsilon}] \bigcup \{h_{0\delta}\}$$

Comment. $P_{0\delta}$ is the intersection of the totality area of the meta-predicate S_H and the meta-predicate R_H. Let us begin the new sub-step of step 0.

- If the requirement is not met, then the step 0 "Forming predicate P_0" ends with the result

$P_0 = \bigcup_{\varepsilon<\delta} P_{0\varepsilon}$ We begin the next step.

Step β: Forming predicate P_β.

- Let us take the element $h_\beta \in \mathrm{Cart}(H)$ and such a natural number m_β, where $h_\beta \in H^{(m_\beta)}$.

- Begin sub-step β0.

Step β: Forming predicate P_β. Sub-step β0.

- Check the requirement $Rel_H(h_\beta)$.

- If the requirement is met, then the "Step β: Forming predicate P_β" continues and we take

$$h_{\beta 0} = h_\beta$$

$$P_{\beta 0} = \{h_{\beta 0}\}.$$

Comment. Since $H \neq \varnothing$, $Rel_H \neq \varnothing$, $Sim_H \neq \varnothing$ and meta-predicate Sim_H are reflexive, then we can say that the set $\{h_{\beta 0}\}$ is the totality area of S_H (because $S_H(h_{\beta 0},h_{\beta 0})$ for each element of set $\{h_{\beta 0}\}$). Since $h_{\beta 0} \in Rel_H$, then we can say that the set $P_{\beta 0}$ is the intersection of some totality area of the meta-predicate Sim_H and the meta-predicate Rel_H. Begin new sub-step of step β.

- If the requirement is not met, then the "Step β: Forming predicate P_β" ends with the result $P_\beta = \varnothing$ and we start a new step.

Step β: Forming predicate P_β. Sub-step βδ.

Let us assume that for each $\varepsilon<\delta$ there is a set $P_{\beta\varepsilon}$ formed and the set $P_{\beta\varepsilon}$ is the intersection of some totality area of the meta-predicate Sim_H and the meta-predicate Rel_H. Let us view the set $H^{(m_\beta)} - \bigcup_{\varepsilon<\delta} P_{\beta\varepsilon}$ and check for the requirement

- $$(\exists h \in H^{(m_\beta)} - \bigcup_{\varepsilon<\delta} P_{\beta\varepsilon})[\mathrm{Rel}_H(h) \,\&\, (\forall h' \in \bigcup_{\varepsilon<\delta} P_{\beta\varepsilon})[\mathrm{Sim}_H(h,h')]]$$

- If the requirement is met, then the "Step β: Forming predicate P_β" continues and we take

$$h_{\beta\delta} = h$$

$$P_{\beta\delta} = [\bigcup_{\varepsilon<\delta} P_{\beta\varepsilon}] \bigcup \{h_{\beta\delta}\}$$

Comment. $P_{\beta\delta}$ is the intersection of some totality area of the meta-predicate Sim_H and the meta-predicate Rel_H. Begin the new sub-state of state β.

- If the requirement is not met, then the "Step β: Forming predicate P_β" ends with the result $P_\beta = \bigcup_{\varepsilon<\delta} P_{\beta\varepsilon}$. We begin the next step.

Each step in the meta-procedure described above ends with some sub-step (because there can be no more than α sub-steps). The result of every step is some property or relation of elements in set H, which is the intersection of the relation meta-predicate R_H and a suitable maximal totality area of the similarity meta-predicate S_H.

Definition 6. Let us say that the signature of system $\langle H;\Sigma\rangle$ is induced by meta-predicates Rel_H and Sim_H, if $\Sigma=\{P_\beta|\ P_\beta$ is a result of the step β of the RS meta-procedure and $\beta<\alpha\}$.

Theorem. (Lorents 1993, 2002) For every non-empty set H and meta-predicates Rel_H and Sim_H there is a system $\langle H;\Sigma\rangle$, where the order of signatures is determined exactly and where the signature is induced by the meta-predicates Rel_H and Sim_H.

Example. Let us view the set H, which consists of the elements 0 and 1. Let us assume that we can sense a relation between only two elements (therefore, we can only identify binary relations, are not able to mine not unary, ternary etc. relations). At the same time we assume that we sense the relation as "all-encompassing" – no matter which ordered pair in set H×H we meet, we still sense that the elements in the first and second positions are related. With *similarity*, however, the situation is a bit more complex – an ordered pair, where the first and second elements are the same (or the pair $\langle b,b\rangle$, where $b\in H$) is similar to all pairs from the set H×H, as well as all other ordered pairs that share the same first or second element.

Let us start the RS-meta-procedure and find the corresponding maximal totality areas. For this we first index all possible ordered pairs (from elements in set H): $h_0=\langle 0,0\rangle$, $h_1=\langle 0,1\rangle$, $h_2=\langle 1,0\rangle$, $h_3=\langle 1,1\rangle$. Let us start by finding "similar companions" to h_0: h_0 and h_1 are similar, since they share the first element, and h_0 and h_2 are similar, since they share the second element. However, h_1 and h_2 are not similar to each other, so we need to discard h_2. Next we view the pair h_3. It turns out to be similar to all pairs already selected. Since there are no more pairs, then we have identified one maximal totality area. Using the same approach from the starting point of h_1, then h_2, and finally h_3 - we find all maximal totality areas. In this case, there are two of them – $\{\langle 0,0\rangle,\langle 0,1\rangle,\langle 1,1\rangle\}$ and $\{\langle 0,0\rangle,\langle 1,0\rangle,\langle 1,1\rangle\}$. It is easy to notice that based on the relation and similarity meta-predicates, we have identified two binary relation between the elements in set $H=\{0,1\}$: $\leq$ and $\geq$. It should be noted that there are other binary relations in this set as well. Unfortunately, the *current* idea of relations and similarity does not allow to mine *those*.

Comment. In an analogous fashion it is possible to mine for relations between any types of objects, including relations between formulas. This should allow us to mine systems of inference rules, if we have previously fixed

on the one hand the idea (Rel) about when we can say that a tuple of formulas is related. In other words, when can a tuple be considered an inference step.

on the other hand the idea (Sim) about when we can say that a tuple of formulas is similar.

Next we will view a specific way for mining single or double predicate inference rules from logic formulas. This mining process is based on the (partial) equality of the constructs found by the DST dialogue system and the SR-meta-procedure.

(III) Mining for inference rules: comparison of DST and the SR-meta-predicate

Logic inference rules are basically relations between a set of logic formulas. If we have a set Φ of logic formulas, then with n premise (where $n\geq 1$) the inference rule is any non-empty n+1-connected set of tuples R, where $R\subseteq\Phi\times\ldots\times\Phi\times\Phi$ and the Cartesian multiplication $\Phi\times\ldots\times\Phi\times\Phi$ contains n+1 factors. The rule R is correct if for every $\langle F_1, \ldots , F_n, F_{n+1}\rangle\in R$ and for every interpretation φ, if $\varphi F_1=1$ and … and $\varphi F_n=1$ it follows that $\varphi F_{n+1}=1$ (see Lorents 2000).

The 15-step procedure used by the DST dialogue system (see section 2) allows to mine single and double premise inference steps that *may exist* in a natural language text. These inference steps were ordered pairs or triples formed from formulas that meet certain requirements. Next we see that the requirements in question basically represent certain meta-predicates Rel_Φ and Sim_Φ, where Φ is the number of formulas extracted by the DST procedure.

Relation. The tuple of formulas $\langle F_1, F_2, F_3\rangle$ is considered an inference step in DST, if

I. at least one of the three formulas shares some common formula parts with the other two formulas.
II. for every interpretation of φ, if $\varphi F_1=1$ and $\varphi F_2=1$, then $\varphi F_3=1$.

The requirements above are suitable for ascertaining relation or, in other words, they give us the meta-predicate Rel_Φ.

Similarity. Two inference steps (which in this case mean the tuples formed by the procedure described in section 2 (specifically, see steps 8 and 9)) are considered similar if:

I. the corresponding tuples are a equally long lists of symbols;
II. the sets formed by the symbols in those lists must be one-to-one;
III. the punctuation symbols (commas, parentheses, etc.) must at the same locations;
IV. the logic operation symbols, quantifiers, etc. must be at the same locations.

It is easy to see that in this case the necessary conditions of *binarity*, *localization by exponent*, *reflexivity and symmetry* are met and similarity is ascertained. Therefore, we can claim that:

Theorem. If the steps and requirements of the procedure DST are met, then the requirements for applying the RS-procedure are also met.

Conclusion. The DST procedure is a realization of RS-procedure in a specific area (mining a system of single or double premise inference rules from natural language texts).

Summary

In this work we compared two procedures for mining systems of logical inference rules from natural language texts (which must first be transformed into so-called logic language). The detailed comparison of the procedures identified that the criteria in dialogue system DST about relations between formulas and the similarity between tuples of formulas meet the requirements of the RS-meta-procedure. This proves DST's applicability for mining single and double premise logical inference rules.

References

[1] Kuratowski, K., Mostowski, A. (1967) *Set theory.* North-Holland Publishing Company. Amsterdam

[2] Lorents, P. (1993*) Algebraic Framework for Knowledge Acquisition,* Stockh, KTH, 1993 (36 pp) (r. report)

[3] Lorents, P. (2000) *Keel ja loogika* (*Language and Logic*) Tln, 2000 (179 pp) (Monograph)

[4] Lorents, P. (2002) *A System Mining Framework.* The 6th World Conference on Systemics, Cybernetics and Informatics, Proceedings Vol I, 195 – 200, Orlando USA, 2002. (pre-referred)

[5] Potter, M. (2004) *Set Theory and its Philosphy.* Oxford University Press.

[6] Matsak E. (2005). Dialogue system for extracting Logic constructions in natural language texts. Proceedings of the International Conference on Artificial Intelligence. IC – AI'2005. Volume II p. 791 – 797. Las Vegas, Nevada, USA

[7] Matsak E. (2006). System DST for Transforming Natural Language Texts, Representing Estimates and Higher Order Predicates and Functionals. The 3rd International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2006. Volume III p. 79 – 84. Orlando, Florida, USA

[8] Matsak E. (2007). The prototype of system for discovering of inference rules. Proceedings of the International Conference on Artificial Intelligence. IC – AI'2007. Volume II p. 489 – 492. Las Vegas, Nevada, USA

[9] Matsak E. (2008). Improved version of the natural language dialog system DST and its application for discovery of logical constructions in children's speech. Proceedings of the International Conference on Artificial Intelligence. IC – AI'2008. Volume II p. 332 – 338. Las Vegas, Nevada, USA

[10] Matsak E. (2009). Discovering Logical Constructs from Estonian Children Language. *Thesis on Informatics and System Engineering.* TUT Press. Estonia, Tallinn.