GAME GRAPH IN THE SYNTACTIC ANALYSIS OF SENTENCES IN KNOWLEDGE MANAGEMENT

Adam DEPTUŁA

Summary: One of the basic skills of artificial intelligence is the ability to communicate in the natural language. By the natural language theory we understand an interdisciplinary stream of research on the colloquial language, the main aim of which is to construct categories used for a detailed description of dependencies and laws governing the natural language. The most important task is to build an adequate grammar of a language describing the mechanism of generating the endless amount of tasks. Nowadays, there are many tools for the syntactic and semantic analysis of the sentences syntax. A possibility of applying game graphs out of Non-deterministic Finite-state Automaton (NFA) in order to describe relations among sentence parts was presented in this paper.

Key words: Natural Language Processing (NLP), game graph, Non-deterministic Finite-state Automaton (NFA), language grammar, knowledge management

1. Introduction

The Peirce's understanding of a sign defines the three-component relation between the material conveyor of meaning, its object and the effect of their connection – the meaning. In a general meaning, the sign is applied in the description of information circulation. If there is no information circulation without signs, then the term of a sign is logically primary [1, 2]. The term of information is widely used in technology – in communication, information technology and linguistics. The amount of information flow, the quality information, structural, syntactic, semantics and pragmatics are taken into account here. Signs are used both in the communication and in getting to know the reality. In a wide scope, the natural language processing is seen as a strictly technical and engineering issue.

Natural languages differ from the formal language of mathematics and from formal languages of programming which have a small, precisely specified amount of rules of creating correct expressions, thanks to which it is possible to construct and recognise correct expressions of a given language [3]. Natural languages are languages used by people for communication. The Natural Language Processing (NLP) is focused on widely understood algorithms operating on these languages. It is most frequently used to extract information but it also has other applications such as voice control, semantic web and automatic translations. Modern linguistics comes from the Noam Chomsky's idea [4]. The beginnings of NLP go back to the 1950s when an artificial intelligence test was proposed by Alan Turing in 1950 [5] and since then it has been a dynamically developing branch of IT and knowledge management.

Papers concerning the subject of a natural language are strictly connected with IT and formal linguistics.

2. Syntactic and semantic analysis of a sentence

The Natural Language Understanding is a "bottleneck", not only for the communication process between a machine and a man but also for the whole artificial intelligence (AI). The polysemy of terms and the dependency of the meaning on the context in a grammatical structure of a sentence is a significant issue for logicians, linguists and AI researchers. The concept of the natural language understanding elaborated within the AI is based on the analysis of the grammatical structure of sentences called a *syntactic analysis* or a *parsing*. Parsing is a process of setting a string of words of the analysed text in a logical structure by means of parenthesis of appropriate phrases [6].

An analysis includes a four-step procedure:

1) a syntactic analysis
2) a semantic interpretation – an analysis of meaning
3) a polysemy analysis – a disambiguation,
4) an incorporation – an analysis of a base of the knowledge system (computer implementation).

The basic idea is based on the construction of sentences according to simple rules in the subsequent stages. When we apply many rules and stages, we obtain many grammatical constructions of sentences. The basic rule of creating a sentence is to connect a noun and a verb phrase (the formal rule formula is as follows: S→NP VP, where NP- a noun phrase, VP- a verb phrase). Phrases can be formed in many different ways both out of the phrases and single words. In the theory of a language, units that are indivisible from the grammatical point of view are known as finite expressions. The expressions belong to such categories as: nouns, verbs, prepositions as well as noun, verb or adjective phrases, etc. A semantic analysis is made in accordance with a grammatical decomposition: the meaning of words, signs and final symbols form possible relations of phrases meanings which make the possible meaning of sentences [6].

3. The natural language grammar

A syntactic analysis of a sentence aims at identification of sentence parts and assigning roles to particular words, with taking into consideration grammar rules of a language and a description (grammar) of how words can be connected to each other in the analysed language. Such actions are essential in order to search and connect ontology with a particular occurrence of terms. The form of rules decides about the power of a given grammar. Chomsky [4] proved that context grammar is essential for the language description. In practise, context-free grammar or link grammar have many applications [7, 8, 9].

3. 1. The link grammar

The link grammar was proposed by Daniel Sleator and Davy Temprley [10]. Each sentence can have many different meanings (as a result of different kinds of polysemy: syntactic, semantic, pragmatic, etc.). However, for particular words and relations among them, the meaning of the whole sentence is unambiguously determined. This kind of grammar is based on the phenomenon called planarity which is a feature of most natural languages. It is based on the fact that if we draw curves between the connected words, they will not intersect. The curves illustrate the relations between a pair of connected words – from connecting words in pairs.

A sequence of words is a sentence of a language described by link grammar if it is possible to draw curves among particular words of a sentence in such a way that the following conditions are met:

a) curves do not intersect (when they are above the words)
b) curves form a consistent graph (that is there is a path from any point of a word in a sequence to any other word in a sequence provided that curves are continuous)
c) requirements of connecting each word present in a sentence are met

In a sentence, for a chosen copula set, all of them have to take part in the construction of curves with other words. Another approach is the categorial grammar invented Kazimierz Ajdukiewicz in 1934 [11] and by Joachim Lambek. Categorial grammar is a formal system which takes into consideration relations among categories (S, NP, VP, TV) by means of function dependencies (likes= (NP $\rightarrow$ S) $\leftarrow$NP). The final transformation leads up to the so-called Ajdukiewicz notation.

3. 2. Tools used in the grammatical analysis of the natural language

Most NLP problems are connected with generation and understanding of sentences e.g. a morphological model of a sentence (the structure of words) which should be built by a computer programme. In this scope, a syntactic analysis (parsing) is of key importance because grammar of a natural language is ambiguous and there are usually many possibilities of a syntactic analysis of one sentence. NLP includes any programmes which can be used to support working with text e.g.: Expert System S.p.A, NLP Software Packages, PSI- Toolkit [12]. Parsers used in the natural language analysis are among others: Link Grammar Parser, Part of Speech Tagging, PCFG's, MiniPar [9]. Other tools include structural techniques. Semantic techniques are an example of techniques which belong to them. The language record is a set of statements and relations of knowledge among them. When using these terms, we can form the so-called "network of statements" – as a graph, the nodes of which are composed of statements and the branches of which are composed of relations. Semantic networks are usually connected with frames or rules. Subsequent tools are banks of trees. Banks of trees are bodies in which each sentence is parsed. The sentence structure is represented in the form of a tree structure.

Signal-flow game graphs from Non-deterministic Finite-state Automatons can be alternative tools in the theory of formal languages [13, 14].

4. Non-deterministic Finite-state Automaton (NFA)

The Non-deterministic Finite-state Automaton (NFA) is a mathematical model processing words from a formal language generated by a regular context-free grammar. The automaton possesses at least one internal state where there are at least two different transitions to the next states for a given input signal.

From a formal point of view, the Non-deterministic Finite-state Automaton is defined in the following way:

$$\langle Z, Q, \gamma, q_0, F \rangle \tag{1}$$

where:

Z- an input alphabet of the automaton, a finite set of input symbols

Z=$\langle z_1, z_2, z_3, ..., z_n \rangle$,

Q- a finite set of internal states of the automaton,

Q=$\langle q_0, q_1, q_2, ..., q_n \rangle$,

γ- the function of the automaton transitions

$\gamma[q(t), z(t)] = \{q_{i_1}(t+1), q_{i_2}(t+1), ..., q_i(t+1)\}$

q_0 - the automaton initial state

F - a set of the automaton final states

NFA with the so-called "empty" transitions, which it is easy to determine on the basis of the context-free grammar rules, is the one that is most frequently used in practice [13, 14]. Game graphs can be a generalisation of NFA in the theory of formal languages and the syntactic analysis of sentences [15, 16, 17].

5. Signal-flow graph (SFG)

Directed graph (similarly to NFA) is defined by an orderly pair of sets. The first of them contains the graph vertex, and the second is composed of the graph arc that is an orderly pair of vertices. The Figure 1 presents an exemplary directed game graph.

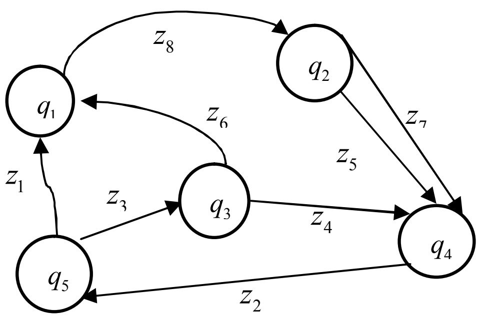


Fig. 1. Directed game graph

Directed game graph in the Figure 1 is composed of the set of Q vertices:

$$Q = \{q_1, q_2, q_3, q_4, q_5\}$$

and of the Z arcs set, that is an orderly pair of vertices:

$$Z = \{z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8\}.$$

The game graph distribution in the Figure 4 goes from the initial vertex q_3 to the expression G_i^+ :

$$G_{q3}^+ = (^0 q_3 (^1 z_4 q_4 (^2 z_2 q_5 (^3 z_1 q_1 (^4 z_8 q_2 (^5 z_5 q_4, z_7 q_4)^5)^4, z_3 q_3)^3)^2, z_6 q_1)^1)^0$$

and then to the expression G_i^{++}:

$$G_{q3}^{++} = ({}^{0} q_3 ({}^{1} z_4 q_4 ({}^{2} z_2 q_5 ({}^{3} z_1 q_1 ({}^{4} z_8 q_2 ({}^{5} z_5 q_4^1, z_7 q_4^2)^5)^4, z_3 q_3^1)^3)^2, \\ z_6 q_1 ({}^{2} z_8 q_2 ({}^{3} z_5 q_4^3, z_7 q_4^4)^3)^2)^1)^0 .$$

The graph distribution goes from the vertex chosen in the first step to the tree structure with cycles and then, to a general game tree structure. Each structure has an appropriate analytic formula G_i^+ and G_i^{++}. A precise algorithm of the graph distribution has been shown in the papers [15, 17].

The Figure 2 shows a game tree structure from the initial vertex q_3.

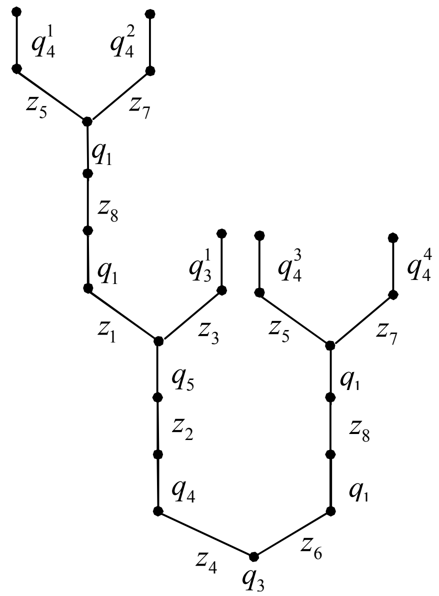
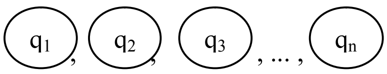


Fig. 2. Game tree structure with the initial vertex q_3

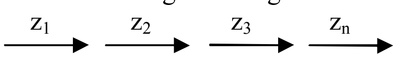
Game graphs can make it possible to make a syntactic analysis (parsing) and can be an alternative method to support computer work with text and process words of the formal language. The following rules are adopted for game graphs structures.

The graph vertex means noun phrases (NP) and verb phrases (VP):



→ NP, VP (nouns, verbs)

Syntactic functions of cases correspond to output arcs. Questions about verb and noun phrases into which a given arc goes are assigned to arcs..



→ questions: what for?, where? ...

Relations among vertices and arcs coming out of them (Figure 3) play a role of conjunctions. In such a case, there are two situations:

a) when only one arc comes out of a vertex, then the following conjunctions are assigned to such a relation: coordinating conjunctions presenting a contrast (e.g..: *but, yet, however*), those presenting non-contrasting ideas (e.g. *and*), those presenting a consequence (e.g. *so*);
b) when more arcs go out of the vertex, then the following conjunctions are assigned to such a relation: correlative conjunctions (e.g. *either ... or, neither ... nor*) and full stops (if the graph describes more than one sentence).

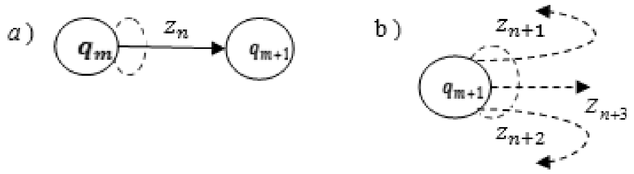


Fig. 3. Relations between vertices and arcs coming out of them

Metaphors cause a lot of problems in the sentence analysis. Not only does the meaning of the sentence depend on the context, but also the structure of the context has an influence on the meaning of the sentences and of the whole text.

A directed game graph ensures an appropriate sentence syntax because arcs coming out of vertices are always taken into consideration in a specific order in the clockwise direction (Figure 4).

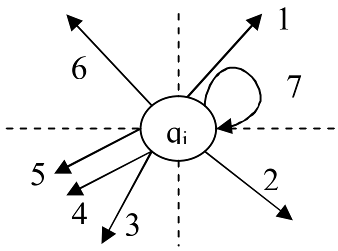


Fig. 4. The order of analysing arcs coming out of a given graph vertex [15, 17]

The Figure 5 shows three sentences written in the form of a game graphs:
Andrew from Opole likes Mary. Andrew likes Mary and misses her very much. Andrew from Opole likes Mary from Cracow.
The Figure 6 shows game structures for graphs shown in the Figure 5 with a semantic interpretation.

$$G^{++}_{1.NP(Andrew)} = ({}^0\,\mathrm{NP(Andrew)}({}^1 \xrightarrow{\text{where?}} \mathrm{NP(Opole)}, \xrightarrow{\text{what she } does?} \mathrm{VP(like)}({}^2 \xrightarrow{\text{whom?}} \mathrm{NP(Mary)}$$
$$({}^3 \xrightarrow{\text{skąd?}} \mathrm{NP(Cracow)})^3)^2)^1)^0$$

$$G^{++}_{2.NP(Andrew)} = ({}^{0}\,\mathrm{NP(Andrew)}({}^{1} \xrightarrow{\text{where?}} \mathrm{VP(Opole)}, \xrightarrow{\text{what she } does?} \mathrm{VP(likes)}({}^{2} \xrightarrow{\text{whom?}} \mathrm{NP(Mary)})^{2})^{1})^{0}$$

$$G^{++}_{3.NP(Andrew)} = ({}^{0}\,\mathrm{NP(Andrew)}({}^{1} \xrightarrow{\text{what she } does?} \mathrm{VP(likes)}({}^{2} \xrightarrow{\text{whom?}} \mathrm{NP(Mary)})^{2}, \xrightarrow{\text{what she } does?} \mathrm{VP(misses)})^{1})^{0}$$

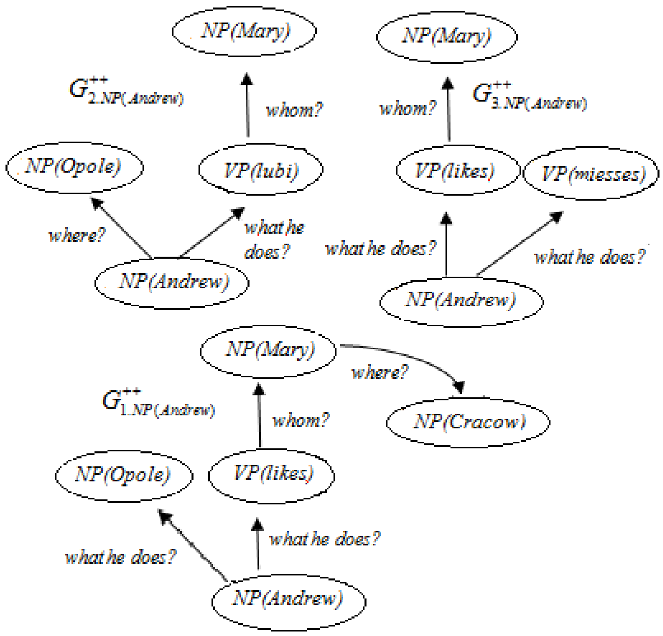


Fig. 5. Game graphs for exemplary sentences

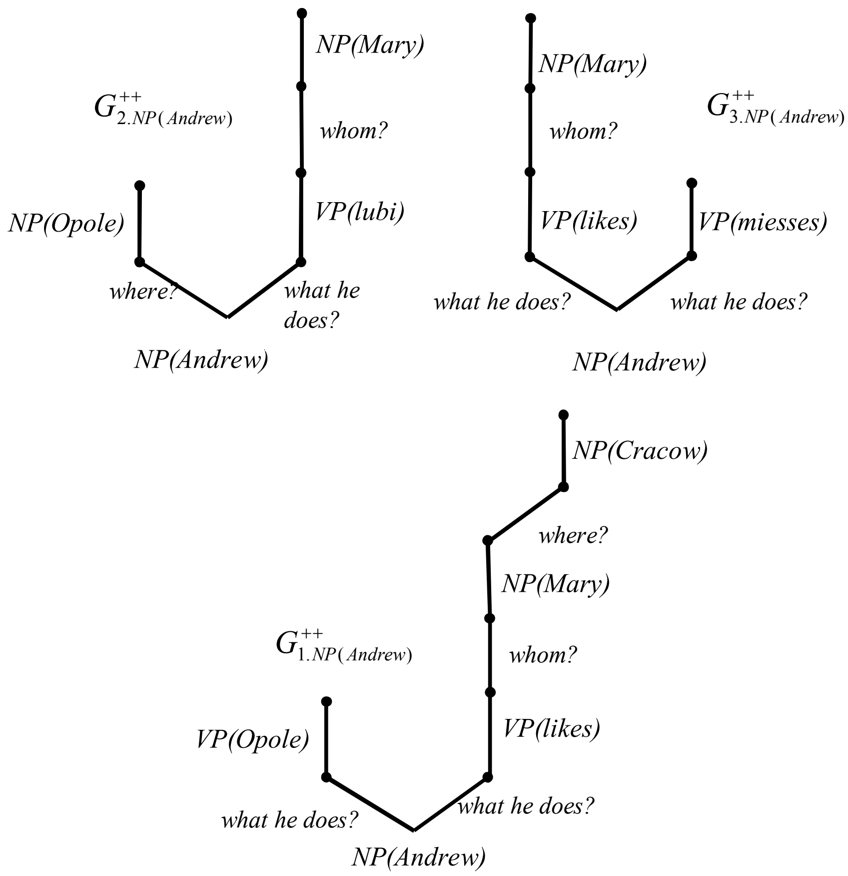


Fig. 6. Game structures for the graphs in the Figure 8

Game graphs also make it possible to write and state the meaning of the sequence of sentences in a correct way, e.g. the first sequence:
John went to Berlin. He signed contracts. He bought flowers for his wife. He came back to Warsaw.
has a different meaning than the second sequence:
John went to Berlin. He signed contracts. He came back to Warsaw. He bought flowers for his wife.
In the first case, we understand that flowers were bought in Berlin, whilst in the second case, we understand that they were bought in Warsaw.
Dependency graphs and game structures for both sequences of sentences were shown in the Figure 7.

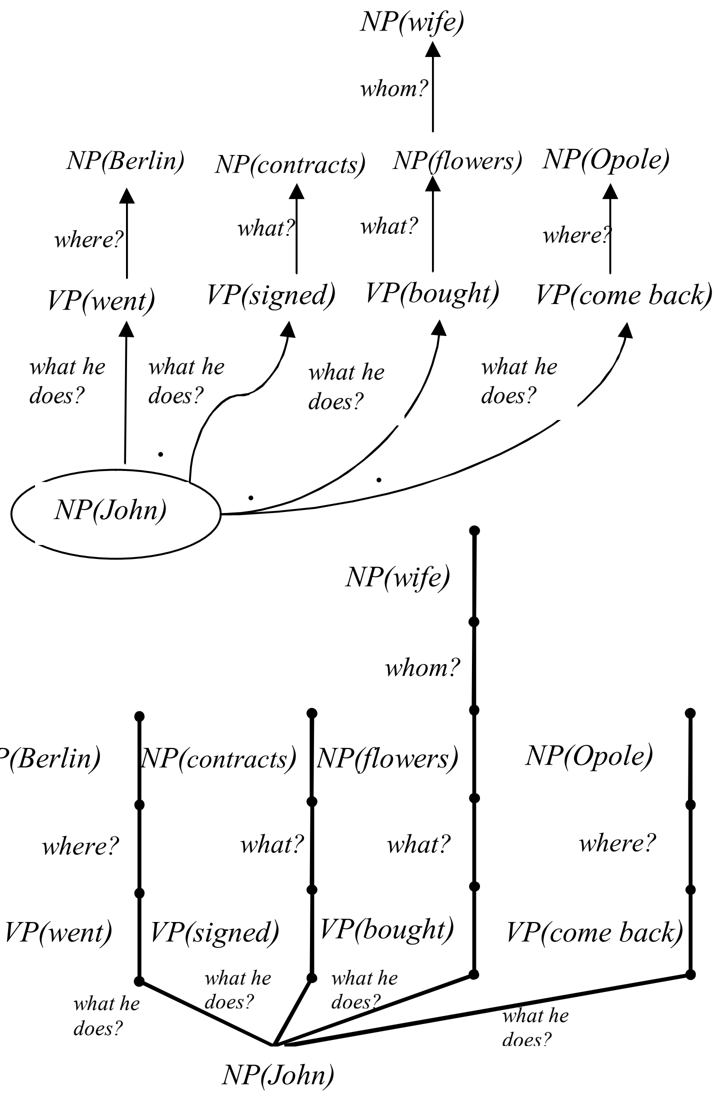


Fig. 7. Dependency graphs and game structures for two sequences of sentences

6. Conclusions

Every grammar should meet two criteria:

- generate all correct sentences of a given language,
- generate only sentences of this language (so it cannot generate incorrect sentences).

Several problems appear when we try to define grammars for a natural language. Among the most important ones it is possible to enumerate among others: discontinuous phrases and sentences, compound sentences, free word order, etc. That is why there are many formalisms for the description of a natural language: transformational grammars, Montague grammars, phrase structures grammars, tree construction grammars, unification grammars. Game graphs can represent systemic functional grammar, referring a linguistic structure to the language (usage) function. In comparison to the context-free grammar and the link grammar, questions about verb and noun phrases are assigned to arcs. Game graphs describe the grammar focused not on particular parts of a sentence but on relations among particular words. Logical operators (and, or, no) can be present between connections after further generalisations and modifications.

References:

1. Atkin, A. Peirce's Theory of Signs, The Stanford Encyclopedia of Philosophy, 2010
2. Kamiński S., Kierunki rozwoju problematyki semiotycznej, 1994.
3. Żegleń U. M., Wprowadzenie do semiotyki teoretycznej i semiotyki kultury,Wydawnictwo Uniwersytetu Mikołaja Kopernika, Toruń 2000.
4. Chomsky N, Zagadnienia teorii składni, Ossolineum 1982.
5. Murawski K., Obliczenia ewolucyjne - geneza i zastosowanie, Biuletyn Instytutu Automatyki i Robotyki WAT, Nr15/2001.
6. Kisielewicz A, Sztuczna inteligencja i logika-podsumowanie przedsięwzięcia naukowego, WNT, 2011.
7. Kazimierczak J., Knowledge reprezentation, Illinois, 1991.
8. Kazimierczak J., Splitting Natural Language Sentences, Wrocław, 1992.
9. http://webdocs.cs.ualberta.ca/~lindek/minipar.html.
10. http://www.link.cs.cmu.edu/link/
11. Ajdukiewicz, K., Język i poznanie t1. Warszawa, PWN, 1960.
12. Charniak, E, Introduction to artificial intelligence, s. 2. Addison-Wesley, 1984.
13. Bromirski J., Teoria automatów, WNT, Warszawa 1971.
14. Kazimierczak J., System cybernetyczny, Wiedza Powszechna, Omega, Warszawa 1978.
15. Kazimierczak J., Teoria gier w cybernetyce, Wiedza Powszechna, Omega, Warszawa 1973.
16. Deptuła A., Partyka M.A., Application of dependence graphs and game trees for decision decomposition for machine systems, Journal of Automation, Mobile Robotics & Intelligent Systems, 2011, vol.5, No.3, pp.17-26.
17. Deptuła A., Partyka M.A, Badanie własności dynamicznych układów maszynowych z uwzględnieniem wielokrotnej numeracji wierzchołkowej dla drzew rozgrywających parametrycznie; Napędy i Sterowanie 3/2010.

MSc Eng. Adam DEPTUŁA
Opole University of Technology
Department of Knowledge Engineering
Faculty of Production Engineering and Logistics
75 Ozimska Street, 45-370 Opole, Poland
phone/fax: (0-77) 449 8733
e-mail: a.deptula@po.opole.pl