Automated Semantic Tagging of Textual Content

Jelena Jovanovic, *University of Belgrade, Serbia*
Ebrahim Bagheri and John Cuzzola, *Ryerson University, Canada*
Dragan Gasevic, *Athabasca University, Canada*
Zoran Jeremic, *Ryerson University, Canada*
Reza Bashash, *SideBuy Technologies, Canada*

Researchers have developed innovative solutions for automated information extraction from text. This article focuses on a subset of such tools—semantic taggers. The authors offer insight into the process and capabilities of semantic tagging and present criteria for choosing a tagger.

In the last few years, there has been a constant increase in the number and variety of online applications that rely on machine comprehension of human language to offer advanced functionalities, such as semantic search, question answering, and recommendation. These functionalities are often enabled by information extraction services that couple text analysis and machine-learning methods and techniques, with large, general-purpose knowledge bases such as Wikipedia.

Information extraction—an active area of text analysis research—has experienced steady growth in recent years.[1] The latest developments in data storage and processing, enabled by cloud infrastructure, have synergized many advanced information extraction methods and techniques, so that text can be processed and relevant information extracted almost instantaneously. The significant increase in the efficiency of automated text analysis, coupled with an increase in the overall quality of extracted information, has made information extraction services appealing for a variety of real-world application areas:

- *Knowledge management*. Information extraction from unstructured text can produce structured, unambiguous (meta)data, thus enabling more effective and efficient search over and management of the organization's content repositories, as exemplified by the KnowMed (http://knowmed.com) solution for healthcare and medical research.
- *Business analytics*. Tools, such as RavenPack News Analytics (www.ravenpack.com/services/

 Published by the IEEE Computer Society 1520-9202/14/$31.00

rpna_dj.htm), analyze news articles to extract diverse kinds of entities and events that could be relevant for business decision making.
- *Social media monitoring and reputation management*. Organizations and individuals can keep track of social media chatter and manage their reputation by leveraging information extraction techniques, as showcased by Radian6 (http://radian6.com).
- *Contextual advertising*. Solutions such as the one developed by ADmantX (www.admantx.com) enable better positioning of advertisements on webpages based on the semantics of the main content of the page—namely, the entities mentioned in the text, the opinions or emotions expressed, and the message that the text aims to communicate.

Here, we focus on a specific information extraction task—the extraction and disambiguation of entities and topics mentioned in or related to a given text. We refer to this task as *semantic tagging* and the tools that perform this task as *semantic taggers*. After providing some insight into the task of semantic tagging and how it is typically performed, we offer a comparative overview of state-of-the-art semantic taggers. We don't aim to provide an exhaustive or in-depth review of semantic tagging techniques and algorithms; rather, our objective is to inform IT practitioners about the potential of semantic taggers. We also discuss practical issues related to the use of these tools, such as the criteria to consider when choosing a tool.

Semantic Tagging Tools

Named entity recognition (NER) is a traditional information extraction task that consists of recognizing entities of a restricted set of types (such as Person, Organization, and Date) in a given text.[2] However, the richness of information that state-of-the-art NER tools provide is rather limited in two significant ways. First, the set of supported entity types is restricted, with the majority of tools only supporting up to a dozen types. Second, although NER tools can recognize that a piece of text represents a certain entity and its type, they cannot determine the "identity" of the entity.

For example, in the sentence, "Novak Djokovic is number one on the ATP list," a NER tool would identify Novak Djokovic as an entity of type Person. However, it would not be able to relate it to the corresponding real-world entity. The term "ATP" would impose an even greater challenge: whereas a NER tool might be able to recognize "ATP" as an entity of type Organization, it would have difficulties relating it with the Association of Tennis Professionals, because this abbreviation can have many different meanings.

The task of relating a piece of text with the actual intended real-world entity is called *disambiguation* or *entity linking*. It consists of relating an entity recognized in the text with an entry in a knowledge base—such as Wikipedia—that uniquely identifies and provides further information about that entity.

Semantic tagging tools can be thought of as an advanced version of NER tools that don't suffer from the deficiencies just mentioned. Semantic tagging is a kind of formal (that is, machine processable) information overlay that has explicitly defined semantics, typically expressed as a reference to an entity or resource defined in a knowledge base or ontology.[3] Semantic taggers overcome the limitations of NER tools. They can recognize all entity types defined in the underlying knowledge base or ontology, which are often on the order of thousands of types, and their disambiguation processes are facilitated by the entities and resources explicitly defined in the underlying knowledge base or ontology.

While the first generation of semantic tagging tools relied primarily on domain-specific ontologies,[3] today's state-of-the-art taggers are based on large, general-purpose, Web-based knowledge bases—primarily, Wikipedia and its more structured and semantics-rich derivatives such as DBpedia (http://dbpedia.org), YAGO (www.mpi-inf.mpg.de/yago-naga/yago), and Freebase (www.freebase.com). The use of these large-scale knowledge bases allows for overcoming the problem of ontology brittleness prevalent in the previous generation of semantic taggers,[4] which caused a decline in their performance (especially recall) if the processed text spanned beyond the domain of the underlying lexicon.

Tool Classifications

Semantic tagging tools can be classified as *manual, automated*, and *semi-automated*.[3] Manual tools require human participation throughout the tagging process. Automated taggers process the text in complete isolation and without any guidance from the user. Semi-automated taggers let users intervene in the tagging process by choosing the best option from the list of candidate tags or by removing some of the proposed tags that they consider incorrect or irrelevant. The dependence on human active participation affects the efficiency of manual and semi-automated tools and makes them less desirable for the kinds of application cases described earlier.

Therefore, in this article, we focus on tools that allow for automated semantic tagging of textual content.

For the sake of brevity, in the rest of the article, we simply refer to this category of tools as *taggers*.

To further distinguish among different kinds of taggers, we use the types of problems and tasks they can deal with as the criteria for differentiation. Similar to other work,[5] we identify the following types of tasks:

- Document topics—identifying topics (that is, concepts from a knowledge base) that are relevant for the overall document.
- Document tagging—identifying entity mentions in the document and linking each mention to the appropriate concept(s) from the knowledge base.
- Suggestion of related topics—identifying topics that aren't directly mentioned in the document but are semantically related to the document's content.
- Role assignment—identifying the role that a topic or a concept plays in the given context. For example, in the sentence, "Add the flavor of a banana to your recipe," "banana" would be disambiguated as a kind of fruit that has the "ingredient" role in the given context. In contrast, in "John threw a banana to the monkey," banana still refers to a fruit, but its role in this context would be "projectile."

In each of these types of tasks, each assigned topic, concept, or role is often associated with a score indicating the likelihood of it being correct.

Typical Semantic Tagging Process

State-of-the-art taggers primarily rely on a combined use of text processing, large-scale knowledge bases, semantic similarity measures, and machine-learning techniques.[5,6] Even though each tool has its own unique approach to performing semantic tagging, some commonalities in the underlying methods of different tools can be observed. In particular, in our analysis of state-of-the-art taggers, we have identified three main phases in a typical tagging process: the detection of entity candidates, disambiguation, and result-set pruning.

Detection

The objective of this phase is twofold: identify "mentions" in the input text, which are the parts of text (single words or phrases) to be tagged and identify a set of candidate entities from a knowledge base for each of the mentions. The detection (or "spotting") of mentions and candidate entities is typically done as a dictionary look-up task. Each tool functions using its own custom dictionary of terms, which are matched against the input text.

The dictionary is typically created by extracting entity labels and descriptions from a specific knowledge base. Some tools enrich each dictionary entry with statistical information, such as the frequency of appearance in the knowledge base, link probability, and co-occurrence rates.

Disambiguation

The purpose of this phase is to select, for each mention spotted in the text, the entities that properly reflect its semantics. The selection is made from among numerous candidate entities identified in the detection phase. Numerous approaches have been proposed for accomplishing this task, which can be classified into the following four generalized groups.[7]

The *popularity-based (mention-entity) prior* approach consists of choosing the most prominent entity for a given piece of text (that is, entity mention).[8] It's simple but can lead to erroneous results, which are often caused by the lack of proper attention to the entities' context and the main theme of the input text. For this reason, most taggers combine this type of approach with other approaches.

Context-based approaches consider the context of the mention to be disambiguated and compare it to the context of the candidate entities. The context of a mention is often modeled through bag-of-words, based on which distance measures—such as cosine similarity measure[9] or Wikipedia links-based measure[8]—can be used to determine the similarity between any two given contexts.

Collective disambiguation consists of jointly disambiguating multiple mentions in the input text. This can be viewed as an extension of the context-based approach in the sense that, apart from considering context similarity scores of each mention-entity pair, the collective disambiguation approach also considers the coherence (semantic relatedness) of the target entities.

The *graph-based* approach is a specific version of the collective disambiguation approach originally proposed by Johannes Hoffart and his colleagues and applied in their AIDA tool.[7] In this approach, mentions and candidate entities are represented as vertices in a graph, while weighted edges between mentions and entities, and weighted edges among entities, denote contextual similarity and coherence, respectively. Disambiguation is performed by identifying a dense subgraph that contains exactly one mention-entity edge for each mention, indicating the most likely meaning for the given mention.

Each of these approaches has its pros and cons and should be applied depending on the type of input text. For instance, context-based approaches are suitable for sufficiently long and relatively clean input texts; however, they tend to produce weaker results for shorter texts such as tweets.[5] Hoffart and his colleagues suggested a procedure for selecting the best disambiguation approach depending on the characteristics of the input text.[7]

Pruning

The objective of this phase is to remove tags that would be of no interest to the user—for example, overly general tags or those that are just marginally related to the main subject of the text. Values of the filtering parameters can be either determined by the tool itself through experiments on different test datasets or set by the users. However, not all tools perform result filtering as a separate task.

For example, some tools (AIDA and Lupedia[12]) perform the final selection of the best mention-entity pairs in the disambiguation phase. In contrast, to meet the requirements of high generality and flexibility, another tool (DBpedia Spotlight) postpones the final selection to the post-disambiguation phase, letting its users fine-tune the selection through a number of configurable parameters.[10]

State-of-the-Art Tools

No tagging tool is a priori better than any other: its suitability depends on many factors such as who will be using it, what it will be used for, or which features are the most important to the user.[4] Therefore, we opted for a descriptive comparison of current tagging tools. Our intention is to familiarize readers with the inner workings of the available tools (see Table 1) and help them choose the tools that best fit their needs (see Table 2). Note that although some tools are open for public use, their implementation and algorithmic details aren't necessarily available.

The first comparison framework (Table 1) is aimed at facilitating the comprehension of tools' similarities and differences in the main phases of the tagging process. Accordingly, it focuses on the "white-box" features of open tools—that is, tools whose underlying approach to semantic tagging is made publicly available (such as in a scientific paper or technical report). Among such tools, we selected those that can be either accessed directly as a Web service or downloaded and run locally. In particular, we look at

- TagMe (http://tagme.di.unipi.it),
- DBpedia Spotlight (http://spotlight.dbpedia.org),
- Wikipedia Miner (http://wikipedia-miner.cms.waikato.ac.nz),
- AIDA (https://github.com/yago-naga/aida),
- Illinois Wikifier (http://cogcomp.cs. illinois.edu/page/software_view/Wikifier),
- LUpedia (http://lupedia.ontotext.com), and
- Denote (http://inextweb.com/ denote_demo).

For each tool, we briefly describe the approach applied in each of the three main phases of the semantic tagging process, and we note the custom-made dictionary or dataset that the tool uses in the detection and disambiguation phases.

The second comparison framework (Table 2) compares "black-box" features of both commercial and open tools. We refer to the features as "black-box" because they don't reveal any information about the inner functioning of the tools; instead, they provide insight into the tools' capabilities, access modes, restrictions on (free)

Table 1. Comparison of "white-box" features of open state-of-the-art semantic tagging tools.

	TagMe[5]	DBpedia Spotlight[10]	Wikipedia Miner[6]
Detection of entity candidates	Dictionary-based (Ancor Dictionary—custom-made, indexed dictionary, based on Wikipedia)	Dictionary-based (Lexicon—custom-made dictionary that associates DBpedia resources with appropriate labels)	Pure text processing (detects all n-grams in the input text and keeps those whose link probability exceeds a low threshold set to discard only nonsense phrases and stop words)
Disambiguation*	Popularity-based prior and collective disambiguation approaches	Context-based approach	Popularity-based prior and collective disambiguation approaches
Pruning the results set	Automated pruning based on the average value of each mention's link probability and the coherence between its candidate tag and the candidate tags of the other mentions in the input text	User-defined pruning criteria, which relies on a number of parameters tunable by the user	Automated pruning using a classifier that relies on a number of features of candidate entities, such as prior probability and relatedness between the entity and its context
Source of entities used in entity detection and disambiguation	Wikipedia-based page catalogue	DBpedia Lexicalization dataset (based on http://wiki.dbpedia.org/Lexicalizations)	Label Vocabulary (a kind of Wikipedia-based dictionary)

** Types of disambiguation approaches are described in the section "Typical Semantic Tagging Process"*

use, and the like. We selected features relevant to identifying the right tool for a specific application case.

In addition to considering open tools, we also included the following commercial tools:

- Alchemy API (www.alchemyapi.com),
- Open Calais (www.opencalais.com),
- Wikimeta (http://wikimeta.com),
- Textwise (http://textwise.com), and
- TextRazor (www.textrazor. com).

Tools and services for the semantic tagging of text are offered by an increasing number of companies, so any attempt at providing an exhaustive list of all available tools is destined to fail. Therefore, we opted for a set of tools that could be considered representative based on the number of users as reported by ProgrammableWeb (www.programmableweb.com) or based on citations in academic publications.

We also opted for tools that provide some level of free program-based access, so interested users can test them. An increasing number of enterprise metadata management solutions offer some form of semantic tagging as one of their features (for example, SmartLogic.com, ContentAnalyst.com, and BasisTech.com). However, because those systems don't meet one or more of our criteria, we don't review them here.

Choosing a Tool

When selecting a tagging tool, you should primarily consider the characteristics of the task for which the tool is intended.[4] Certain features of the tagging task are especially important.

Specificity of Subject Domain

The first feature to consider is the specificity of the topics covered by the documents processed. The content can be domain specific (that is, focused on one particular domain, such as medicine or law) or more broad and general (covering a wide range of topics of general interest).

Due to their reliance on general knowledge bases (such as Wikipedia, DBpedia, and Freebase), a

AIDA[7]	Illinois Wikifier[11]	LUpedia[12]	Denote[13]
Pure text processing for entity spotting—a named entity recognition (NER) tool detects noun phrases as candidate entity mentions; YAGO2 is then used to select candidate entities for each mention	Pure text processing for entity spotting; Wikipedia-based anchor-title index is then used to select candidate entities for each mention	Dictionary-based: sequences of tokens from the input text are searched for in a custom-made dictionary (Alias Dictionary) that relates entities with their labels and types	Multiple strategies based on text processing and statistics: inverse document frequency, NER, sentence chunking, N-grams, and nearest neighbor
Popularity-based prior, context-based approach, and graph-based approach	Popularity-based prior, context-based approach, and collective disambiguation approach	N/A	Context-based approach—the best mention is selected through the conditional likelihood of a mention, given the context's subject category
N/A	N/A	Automated pruning based on weights that reflect the level of matching between the entity's alias and the input text, the general "relevancy" of the predicate and the class of the entity	Partially automated and partially user-defined pruning criteria
YAGO2, a general-purpose knowledge base	Wikipedia-based anchor-title index	Alias Dictionary—each alias relates to the corresponding entity, its type, and the predicate defining the alias-entity relationship (derived from DBpedia and http://linkedmdb.org)	A subset of DBPedia datasets.

large majority of the taggers discussed in this article are better suited for general content.

Characteristics of the Text to Be Tagged

It's also important to consider the characteristics of the documents and text to be tagged, such as the length, writing style, and use of jargon (scientific papers versus news articles versus tweets or status updates, for example).

The majority of existing taggers don't specify the type of text they are intended for and claim that, if appropriately configured, they can cover different forms of textual content. Still, some tools specifically target a certain category of text. For example, TagMe is well suited for short texts, such as tweets, status updates, and search result snippets, whereas OpenCalais was developed primarily for news articles.

Response Time Requirements

Another feature to consider is the response time requirements of the application domain—that is, whether the task requires real-time or offline (asynchronous) tagging could be an acceptable alternative. This aspect of the tagging task relates to the efficiency issues that stem from using large-scale knowledge bases in the tagging process. By affecting the speed of the tagger, the use of large knowledge bases (indirectly) affects users' satisfaction with the tool—and eventually their acceptance and adoption of it.

The processing speed might not be a factor if real-time results aren't required—if the task is to develop a Web crawler that would index and semantically tag visited webpages, for example. However, if a usage scenario requires real-time results, you should look for semantic taggers that are efficient even in real time. One strategy, applied in TagMe, is to make a tradeoff between performance and speed, achieving higher speeds by sacrificing slight performance gains offered by sophisticated but computationally more intensive techniques.

Tool Customization

The final feature to consider is the customizability of a tagging tool—whether the tool provides sufficient means to adapt its tunable parameters.

Table 2. Comparison of "black-box" features of state-of-the-art semantic tagging tools.

	TagMe	DBpedia Spotlight	Wikipedia Miner	AIDA	Illinois Wikifier	LUpedia
Supported tagging tasks*	Document tagging (with relevance scores)—DT(RS)	DT(RS)	DT(RS); document topics	DT(RS)	DT(RS)	DT(RS)
Knowledge base(s) for entity disambiguation	Wikipedia	DBpedia	Wikipedia	YAGO2	Wikipedia	DBpedia, Linked Movie Database
Restrictions on free use	None	None	None	None	None	None
Forms of program-based access	RESTful API	RESTful API	RESTful API; Java toolkit	Java toolkit	Java toolkit	RESTful API
Support for entity typing	Wikipedia categories	DBpedai/ Wikipedia types/ categories	No direct support	YAGO types	Wikipedia categories	DBpedia types
**Type of text for which the tool is suitable ** **	Short texts like tweets and search snippets	Longer texts like Web pages or Web feeds	Equally suitable for all kinds of text	Bioinformatics documents and workflows	Short newswire articles	Entertainment and TV related texts

** See the "Tools Classifications" section in the main text*

*** This information hasn't been explicitly made available by the tool developers; it has been implied by other community developers who have used the tools. For a more reliable conclusion, a systematic study is needed.*

Related Work in *IT Pro*

- S. Anderson and K. Mohan, "Social Networking in Knowledge Management," *IT Professional*, vol. 13, no. 4, 2011, pp. 24–28.
- S. Murugesan et al., "The Future of Web Apps," *IT Professional*, vol. 13, no. 5, 2011, pp. 12–14.
- S. Mithas et al., "Leveraging Big Data and Business Analytics," *IT Professional*, vol. 15, no. 6, 2013, pp. 18–20.
- K. Christidis, G. Mentzas, and D. Apostolou, "Supercharging Enterprise 2.0," *IT Professional*, vol. 13, no. 4, 2011, pp. 29–35.
- E.M. Stroz and D.F. Hsu, "Justice Informatics: Improving Knowledge Discovery for the Justice System," *IT Professional*, vol. 14, no. 5, 2012, pp. 47–52.

A tagger's default configuration is often only suitable for some undemanding tagging tasks. Accordingly, a tagging tool tends to be more useful if it lets users define a custom configuration matching the specificities of the task at hand. The main challenge here lies in letting users tune the tagger with respect to the key issues, such as specificity and comprehensiveness, without being concerned with the details of the tool's parameters and internal functioning.

Many of today's tagging tools tend to mitigate this challenge by grouping related parameters and exposing only a couple of generalized and intuitively named parameters (such as confidence). This way, the tools make a tradeoff between advantages obtainable from highly customized tagging engines and the erroneous results that might follow from inappropriately set parameter values.

Denote	Alchemy API	Open Calais	Wikimeta	Textwise	TextRazor
DT(RS); suggest related topics (with RS); role assignment	Document topics (with RS); DT(RS)	Document topics; DT(RS)	DT(RS)	DT(RS)	Document topics (with RS); DT(RS)
DBpedia	DBpedia, Freebase, YAGO, GeoNames, and others (http://lod-cloud.net)	DBpedia, Freebase, Geonames, Wikipedia, Linked Movie Database	DBpedia	N/A	DBpedia, Freebase
500 free API calls/month	30,000 free API calls/day for academic use	50,000 free API calls/day	None for students or 100 free API calls/day	Unspecified no. of free API calls (at TextWise's discretion)	500 free API calls/day
RESTful API	RESTful API; toolkits for major programming languages	RESTful and SOAP Web service API	RESTful API	RESTful API	RESTful API; Python client
DBpedia/Wikipedia types/categories	Hundreds of entity types from a custom classification scheme	39 custom-defined entity types; 21 disambiguated to LOD knowledge bases	Traditional Named Entity types: such as Person, Organization, and Product	No support; a modified version of the Open Directory Project classification scheme is used for doc. topics	DBpedia and Freebase types
Long descriptive documents	News articles, blog posts	News articles, blog posts	Equally suitable for all kinds of text	Equally suitable for all kinds of text	Legal documents, news articles, emails

Automated semantic tagging technology facilitates a deeper analysis of the significant amount of text that is generated worldwide, either in the form of user-generated content (such as tweets and blogs) or enterprise-specific content (such as corporate documents and reports). This technology could allow for more accurate and semantics-aware organization, search, and retrieval of textual information. However, we also need to highlight the importance of choosing the most suitable semantic tagger (or a combination of taggers) for a target application domain by considering factors such as those just outlined. The deployment of suitable automated semantic tagging technology can significantly enhance textual processing by moving beyond syntactical information and into the realm of deeper textual semantics. IT

References

1. S. Feldman, *The Answer Machine: Synthesis Lectures on Information Concepts, Retrieval, and Services*, Morgan & Claypool Publishers, 2012.
2. G.S. Ingersoll, T.S. Morton, and A.L. Farris, *Taming Text—How to Find, Organize, and Manipulate It*, Manning Publications, 2013.
3. V. Uren et al., "Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art," *J. Web Semantics*, vol. 4, no. 1, 2005, pp. 14–28.
4. D. Maynard, "Benchmarking Textual Annotation Tools for the Semantic Web," *Proc. 6th Int'l Conf. Language Resources and Evaluation*, 2008; http://gate.ac.uk/sale/lrec2008/benchmarking.pdf.
5. M. Cornolti, P. Ferragina, and M. Ciaramita, "A Framework for Benchmarking Entity-Annotation Systems," *Proc. 22nd Int'l World Wide Web Conf.*, 2013, pp. 249–260.
6. D. Milne and I.H. Witten, "An Open-Source Toolkit for Mining Wikipedia," *J. Artificial Intelligence*, Jan. 2013, pp. 222–239.

7. J. Hoffart et al., "Robust Disambiguation of Named Entities in Text," *Proc. Conf. Empirical Methods in Natural Language Processing*, 2011, pp. 782–792.
8. D. Milne and I.H. Witten, "Learning to Link with Wikipedia," *Proc. 17th ACM Conf. Information and Knowledge Management*, 2008, pp. 509–518.
9. R. Feldman and J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, Cambridge Univ. Press, 2006.
10. P.N. Mendes et al., "DBpedia Spotlight: Shedding Light on the Web of Documents," *Proc. 7th Int'l Conf. Semantic Systems*, 2011, pp. 1–8.
11. L. Ratinov et al., "Local and Global Algorithms for Disambiguation to Wikipedia," *Proc. 49th Annual Meeting of the Assoc. Computational Linguistics: Human Language Technologies*, 2011, pp.1375–1384.
12. P. Mihaylov, "D4.5 Integration of Advanced Modules in the Annotation Framework," Deliverable of the NoTube FP7 EU project (project no. 231761), 2011; http://notube3.files.wordpress.com/2012/01/notube_d4-5-integration-of-advanced-modules-in-annotation-framework-vm33.pdf.
13. J. Cuzzola et al., "Semantic Tagging with Linked Open Data," Proc. 4th Canadian Semantic Web Symp., 2013; http://ceur-ws.org/Vol-1054/paper-13.pdf.

***Jelena Jovanovic** is an associate professor at the University of Belgrade. Her research interests include semantic technologies and technology enhanced learning. She can be reached at http://jelenajovanovic.net.*

***Ebrahim Bagheri** is an assistant professor at Ryerson University, where he leads the Laboratory for Systems, Software and Semantics (LS3). His research interests are large-scale software reuse and Semantic Web technologies. Bagheri is a senior member of IEEE and a Professional Engineer in the Province of Ontario. Contact him at bagheri@ryerson.ca.*

***John Cuzzola** is a research associate at Ryerson University and lead developer of the Denote semantic annotator. His research interests are in the areas of artificial intelligence, natural-language processing, and machine learning. Contact him at jcuzzola@sd73.bc.ca.*

***Dragan Gasevic** is a professor and Canada Research Chair of semantic and learning technologies at Athabasca University. He is President Elect of the Society for Learning Analytics Research and the Founder of Pro Solo Technologies, which develops a software solution for tracking, evaluating, and recognizing competences gained through self-directed learning and social interactions. His research interests are at the intersect of computational, socio-cognitive, and design aspects of information seeking, sensemaking, and learning. He can be reached at http://dgasevic.athabascau.ca.*

***Zoran Jeremic** is a research fellow at Ryerson University and Senior System Analyst at Athabasca University. His broad research interest is in the domain of technology enhanced learning. He can be reached at http://zoranjeremic.org.*

***Reza Bashash** is CTO and founder of Sidebuy Technologies, Canada. His areas of interest are artificial intelligence and search technologies. Contact him at bashash@gmail.com.*