Open Book: a tool for helping ASD users' semantic comprehension

Eduard Barbu
University of Jaén
Paraje de Las Lagunillas
Jaén, 23071, Spain
ebarbu@ujaen.es

Maria Teresa Martín-Valdivia
University of Jaén
Paraje de Las Lagunillas
Jaén, 23071, Spain
maite@ujaen.es

Luis Alfonso Ureña-López
University of Jaén
Paraje de Las Lagunillas
Jaén, 23071, Spain
laurena@ujaen.es

Abstract

Persons affected by Autism Spectrum Disorders (ASD) present impairments in social interaction. A significant percentile of them have inadequate reading comprehension skills. In the ongoing FIRST project we build a multilingual tool called Open Book that helps the ASD people to better understand the texts. The tool applies a series of automatic transformations to user documents to identify and remove the reading obstacles to comprehension. We focus on three semantic components: an Image component that retrieves images for the concepts in the text, an idiom detection component and a topic model component. Moreover, we present the personalization component that adapts the system output to user preferences.

1 Introduction

Autism Spectrum Disorders are widespread and affect every 6 people in 10000 according to Autism Europe site[1]. The disorder is chiefly characterized by impairments in social interaction and by repetitive and stereotyped behaviour (Attwood, 2007). People affected by ASD are not able to communicate properly because they lack an adequate theory of mind (Baron-Cohen, 2001). Therefore, they are not able to infer the other persons' mental states: beliefs, emotions or desires. This lack of empathy prevents the people with ASD to have a fulfilled social life. Their inability to understand others leads to the incapacity to communicate their wishes and desires and to social marginalization.

[1]http://www.autismeurope.org/

The FIRST project seeks to make a small step towards integration of ASD people in the information society by addressing their reading comprehension ability. It is well known that many of the ASD people have a wide range of language difficulties. Psychological studies showed that they have problems understanding less common words (Gillispie, 2008), have difficulty comprehending polysemous words (Fossett and Mirenda, 2006) and have troubles dealing with figurative language (Douglas et al., 2011). The absence of good comprehension skills impedes the ASD students to participate in curriculum activities or to properly interact with their colleagues in chats or blogs. To enhance the reading comprehension of ASD people we are developing a software tool. It is built by partners in Academia and Industry in close collaboration with teams of psychologists and clinicians. It operates in a multilingual setting and is able to process texts in English, Spanish and Bulgarian languages. Based on literature research and on a series of studies performed in the United Kingdom, Spain and Bulgaria with a variety of autistic patients ranging from children to adults the psychologists identified a series of obstacles in reading comprehensions that the tool should remove. From a linguistic point of view they can be classified in syntactic obstacles (difficulty in processing relative clauses, for example) and semantic obstacles (difficulty in understanding rare or specialized terms or in comprehension of idioms, for example). The tool applies a series of automatic transformations to user documents to identify and remove the reading obstacles to comprehension. It also assists the carers , persons that assist the ASD people in every day life tasks, to correct the results of auto-

11

matic processing and prepare the documents for the users. This paper will focus on three essential software components related to semantic processing: a software component that adds images to concepts in the text, a software component that identifies idiomatic expressions and a component that computes the topics of the document. Moreover, we present the personalization component that adapts the system output to user preferences. The rest of the paper has the following structure: the next section briefly presents other similar tools on the market. Section 3 presents a simple procedure for identifying the obstacles ASD people have in reading comprehensions. Section 4 shows the architecture of the semantic processing components and the personalization component. The last section draws the conclusions and comments on the future work. Before presenting the main part of the article we make a brief note: throughout the paper we will use whenever possible the term "user" instead of ASD people or patients.

2   Related Work

A number of software tools were developed to support the learning of ASD people. Probably the most known one is Mind Reading[2], a tool that teaches human emotions using a library of 412 basic human emotions illustrated by images and video. Other well known software is VAST-Autism[3], a tool that supports the understanding of linguistic units: words, phrase and sentences by combining spoken language and images. "Stories about me" is an IPad application[4] that allows early learners to compose stories about themselves. All these tools and others from the same category are complementary to Open Book. However, they are restricted to pre-stored texts and not able to accommodate new pieces of information. The main characteristics that sets aside our tool is its scalability and the fact that it is the only tool that uses NLP techniques to enhance text comprehension. Even if the carers correct the automatic processing output, part of their work is automatized.

3   Obstacles in text comprehension

Most of the automatic operations executed by the Open Book tool are actually manually performed by the carers. They simplify the parts of the text that are difficult to understand. We compared the texts before and after the manual simplification process and registered the main operations. The main simplification operations ordered by frequency performed by carers for 25 Spanish documents belonging to different genders: rent contracts, newspaper articles, children literature, health care advices, are the following:

1. Synonymous (64 Operations). A noun or an adjective is replaced by its less complex synonym.

2. Sentence Splitting (40 Operations). A long sentence is split in shorter sentences or in a bullet list.

3. Definition (34 Operations). A difficult term is explained using Wikipedia or a dictionary.

4. Near Synonymous (33 Operations). The term is replaced by a near synonym.

5. Image (27 Operations) A concept is illustrated by an image.

6. Explanation (24 Operations). A sentence is rewritten using different words.

7. Deletion (17 Operations). Parts of the sentence are removed.

8. Coreference(17 Operations). A coreference resolution is performed.

9. Syntactic Operation (9 Operations). A transformation on the syntactic parse trees is performed.

10. Figurative Language (9 Operations). An idiom or metaphor is explained.

11. Summarization (3 Operations). The content of a sentence or paragraph is summarized.

The most frequent operations with the exception of Sentence Splitting are semantic in nature: replacing a word with a synonym, defining the difficult

terms. The only obstacle that cannot be tackled automatically is Explanation. The Explanation entails interpretation of the sentence or paragraph and cannot be reduced to simpler operations.

A similar inventory has been done in English. Here the most frequent operation are Sentence Splitting, Synonyms and Definition. The operations are similar across English and Spanish but their ordering differs slightly.

4   The Semantic System

In this paper we focus on three semantic components meant to augment the reading experience of the users. The components enhance the meaning of documents assigning images to the representative and difficult concepts, detecting and explaining the idiomatic expressions or computing the topics to which the documents belong.

In addition to these components we present another component called Personalization. Strictly speaking, the personalization is not related to semantic processing per se but, nevertheless, it has an important role in the final system. Its role is to aggregate the output of all software components,including the three ones mentioned above, and adapt it according to user's needs.

All the input and output documents handled by NLP components are GATE (Cunningham et al., 2011) documents. There are three reasons why GATE documents are preferred: reusability, extensibility and flexibility. A GATE document is reusable because there are many software components developed both in academy and industry, most of them collected in repositories by University of Sheffield, that work with this format. A GATE document is extensible because new components can add their annotations without modifying previous annotations or the content of the document. Moreover, in case there is no dependence between the software components the annotations can be added in parallel. Finally, a GATE document is flexible because it allows the creation of various personalization workflows based on the specified attributes of the annotations. The GATE document format is inspired by TIPSTER architecture design[5] and contains in addition to the text or multimedia content annotations

---

[5]http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/

grouped in Annotation Sets and features. The GATE format requires that an annotation has the following mandatory features: an id, a type and a span. The span defines the starting and the ending offsets of the annotation in the document text.

Each developed software component adds its annotations in separate name annotation sets. The components are distributed and exposed to the outside world as SOAP web services. Throughout the rest of the paper we will use interchangeably the terms: component, software component and web service.

For each semantic component we discuss:

- The reasons for its development. In general, there are two reasons for the development of a certain software component: previous studies in the literature and studies performed by our psychologists and clinicians. In this paper we will give only motivations from previous studies because the discussion of our clinicians and psychologist studies are beyond the purpose of this paper.

- Its architecture. We present both the foreseen characteristics of the component and what was actually achieved at this stage but we focus on the latter.

- The annotations it added. We discuss all the features of the annotations added by each component.

4.1   The Image Web Service

In her landmark book, "Thinking in Pictures: My Life with Autism", Temple Grandin (1996), a scientist affected by ASD, gives an inside testimony for the importance of pictures in the life of ASD people:

"Growing up, I learned to convert abstract ideas into pictures as a way to understand them. I visualized concepts such as peace or honesty with symbolic images. I thought of peace as a dove, an Indian peace pipe, or TV or newsreel footage of the signing of a peace agreement. Honesty was represented by an image of placing one's hand on the Bible in court. A news report describing a person returning a wallet with all the money in it provided a picture of honest behavior."

13

Grandin suggests that not only the ASD people need images to understand abstract concepts but that most of their thought process is visual. Other studies document the importance of images in ASD: Kana and colleagues (2006) show that the ASD people use mental imagery even for comprehension of low imagery sentences. In an autobiographic study Grandin (2009) narrates that she uses language to retrieve pictures from the memory in a way similar to an image retrieval system.

The image component assigns images to concepts in the text and to concepts summarizing the meaning of the paragraphs or the meaning of the whole document. Currently we are able to assign images to the concepts in the text and to the topics computed for the document. Before retrieving the images from the database we need a procedure for identifying the difficult concepts. The research literature helps with this task, too. It says that our users have difficulty understanding less common words (Lopez and Leekam, 2003) and that they need word disambiguation (Fossett and Mirenda, 2006).

From an architectural point of view the Image Web Service incorporates three independent sub-components:

- **Document Indexing**. The Document Indexing sub-component indexes the document content for fast access and stores all offsets of the indexing units. The indexed textual units are words or combinations of words (e.g., terms).

- **Difficult Concepts Detection**. The difficult concepts are words or terms (e.g. named entities) disambiguated against comprehensive resources: like Wordnet and Wikipedia. This sub-component formalizes the notion "difficult to understand" for the users. It should be based on statistical procedures for identifying rare terms as well as on heuristics for evaluating the term complexity from a phonological point of view. For the time being the sub-component searches in the document a precompiled list of terms.

- **Image Retrieval**. This sub-component retrieves the images corresponding to difficult concepts from image databases or from web searching engines like Google and Bing.

The Image Web Service operates in automated mode or in on-demand mode. In the automated mode a document received by the Image Web Service is processed according to the working flow in Figure 1. In the on-demand mode the user highlights the concepts (s)he considers difficult and the web service retrieves the corresponding image or set of images. The difference between the two modes of operations is that in the on-demand mode the difficult concept detection is performed manually.

Once the GATE document is received by the system it is tokenized, POS (Part of Speech) tagged and lemmatized (if these operations were not already performed by other component) by a layer that is not presented in Figure 1. Subsequently, the document content is indexed by Document Indexing subcomponent. For the time being the terms of the document are disambiguated against Wordnet. The Image Retrieval component retrieves the corresponding images from the image database.

The current version uses the ImageNet Database (Deng et al., 2009) as image database. The ImageNet database pairs the synsets in Princeton Wordnet with images automatically retrieved from Web and cleaned with the aid of Mechanical Turk. Because the wordnets for Spanish and Bulgarian are either small or not publicly available future versions of the Web Service will disambiguate the terms against Wikipedia articles and retrieve the image illustrating the article title. All annotations are added in "ImageAnnotationSet". An annotation contains the following features:

- *Image Disambiguation Confidence* is the confidence of the WSD (Word Sense Disambiguation) algorithm in disambiguating a concept.

- *Image URL* represents the URL address of the retrieved image

- *Image Retrieval Confidence* is the confidence of assigning an image to a disambiguated concept.

In the on-demand mode the images are also retrieved from Google and Bing Web Services and the list of retrieved images is presented to the carer and/or to the users. The carer or user selects the image and inserts it in the appropriate place in the document.
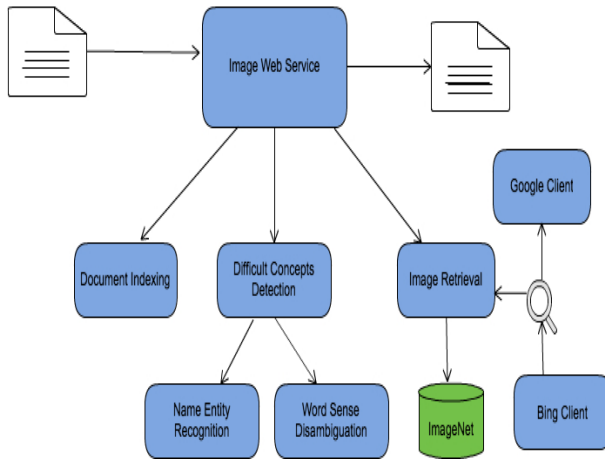
Figure 1: The Image Web Service.

4.2 The Idiom Detection Web Service

In the actual linguistic discourse and lexicographical practice the term "idiom" is applied to a fuzzy category defined by prototypical examples: "kick the bucket", "keep tabs on", etc. Because we cannot provide definitions for idioms we venture to specify three important properties that characterize them (Nunberg et al., 1994) :

- Conventionality.The meaning of idioms are not compositional.

- Inflexibility. Idioms appear in a limited range of syntactic constructions.

- Figuration. The line between idioms and other figurative language is somewhat blurred because other figurative constructions like metaphors: "take the bull by the horns" or hyperboles: "not worth the paper it's printed on" are also considered idioms.

The figurative language in general and the idioms in particular present particular problems for our users as they are not able to grasp the meaning of these expressions (Douglas et al., 2011). To facilitate the understanding of idiomatic expressions our system identifies the expressions and provide definitions for them.

The actual Idiom Web Service finds idiomatic expressions in the user submitted documents by simple text matching. The final version of Idiom Web Service will use a combination of trained models and hand written rules for idiom detection. Moreover, it is also envisaged that other types of figurative language like metaphors could be detected. At the moment the detection is based on precompiled lists of idioms and their definitions. Because the component works by simple text matching, it is language independent. Unlike the actual version of the Idiom Web Service the final version should be both language and domain dependent. The architecture of this simple component is presented in Figure 2 .
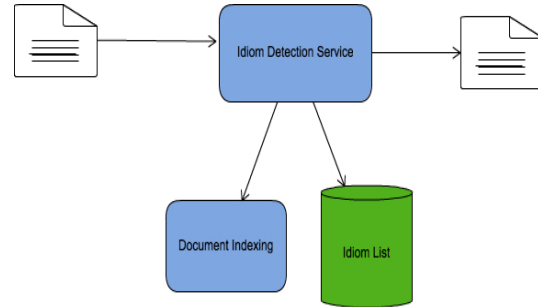

Figure 2: The Idiom Web Service.

The GATE input document is indexed by the document indexing component for providing fast access to its content. For each language we compiled list of idioms from web sources, dictionaries and Wikipedia. All idiom annotations are added in the "IdiomAnnotationSet". An annotation contains the following features:

- *Idiom Confidence* represents the confidence the algorithm assigns to a particular idiom detection.

- *Definition* represents the definition for the extracted idiom.

4.3 The Topic Models Web Service

The mathematical details of the topics models are somewhat harder to grasp but the main intuition behind is easily understood. Consider an astrobiology document. Most likely it will talk about at least three topics: biology, computer models of life and astronomy. It will contain words like: cell, molecules, life related to the biology topic; model, computer, data, number related to computer models of life topic and star, galaxy, universe, cluster related with astronomy topic. The topic models are used to organize vast

collections of documents based on the themes or discourses that permeate the collection. From a practical point of view the topics can be viewed as clusters of words (those related to the three topics in the example above are good examples) that frequently co-occur in the collection. The main assumption behind Latent Dirichlet Allocation (LDA) (Blei et al., 2003), the simplest topic model technique, is that the documents in the collections were generated by a random process in which the topics are drawn from a given distribution of topics and words are drawn from the topics themselves. The task of LDA and other probabilistic topic models is to construct the topic distribution and the topics (which are basically probability distributions over words) starting with the documents in the collection.

The Topic Models Web Service is based on an implementation of LDA. It assigns topics to the user submitted documents, thus informing about the themes traversing the documents and facilitating the browsing of the document repository. The topics themselves perform a kind of summarization of documents showing, before actual reading experience, what the document is about.

The architecture of the Topic Models Web Service is presented in Figure 3.
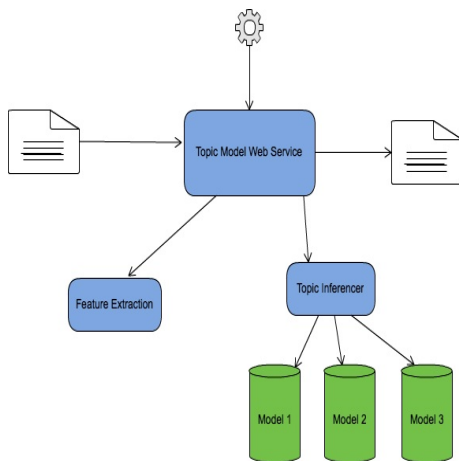

Figure 3: The Topic Model Web Service.

Once a document is received it is first dispatched to the Feature Extraction Module where it is POS tagged and lemmatized and the relevant features are extracted. As for training models, the features are all nouns, name entities and verbs in the document. Then the Topic Inferencer module loads the appropriate domain model and performs the inference and assigns the new topics to the document. There are three domains/genders that the users of our system are mainly interested in: News, Health Domain and Literature. For each of these domains we train topic models in each of the three languages of the project. Of course the system is easily extensible to other domains. Adding a new model is simply a matter of loading it in the system and modifying a configuration file.

The output of the Web System is a document in the GATE format containing the most important topics and the most significant words in the topics. The last two parameters can be configured (by default they are set to 3 and 5 respectively). Unlike the annotations for the previous components the annotation for Topic Model Web Service are not added for span of texts in the original document. This is because the topics are not necessarily words belonging to the original document. Strictly speaking the topics are attributes of the original document and therefore they are added in the "GateDocumentFeatures" section. An example of an output document containing the section corresponding to the document topics is given in Figure 4.

```
<GateDocumentFeatures>
<Feature>
  <Name className="java.lang.String">Topic-230</Name>
  <Value className="java.lang.String">contrato,concurso,empresa,acreedor,adjudicación</Valu
</Feature>
<Feature>
  <Name className="java.lang.String">Topic-12</Name>
  <Value className="java.lang.String">banco,deuda,entidad,mercado,liquidez</Value>
</Feature>
<Feature>
  <Name className="java.lang.String">Topic-61</Name>
  <Value className="java.lang.String">grupo,compañía,repsol,tener,participación</Value>
</Feature>
</GateDocumentFeatures>
```

Figure 4: The GATE Document Representation of the Computed Topic Model.

Currently we trained three topic models corresponding to the three above mentioned domains/genres for the Spanish language:

- News. The corpus of news contains more than 500.000 documents downloaded from the web pages of the main Spanish newspapers (El Mundo, El Pais, La Razon, etc. . .). The topic model is trained using a subset of 50.000 documents and 400 topics. The optimum number of documents and topics will be determined when

the users test the component. However, one constraint on the number of documents to use for model training is the time required to perform the inference: if the stored model is too big then the inference time can exceed the time limit the users expect.

- Health Domain. The corpus contains 7168 Spanish documents about general health issues (healthy alimentation, description of the causes and treatments of common diseases, etc.) downloaded from medlineplus portal. The topic model is trained with all documents and 100 topics. In the future we will extend both the corpus and the topic model.

- Literature. The corpus contains literature in two genders: children literature (121 Spanish translation of Grimm brothers stories) and 336 Spanish novels. Since for the time being the corpus is quite small we train a topic model with 20 topics just for the system testing purposes.

For the English and the Bulgarian language we have prepared corpora for each domain but we have not trained a topic model yet. To create the training model all corpora should be POS tagged, lemmatized and the name entities recognized. The features for training the topic model are all nouns, name entities and verbs in the corpora.

4.4 Personalization

The role of the Personalization Web Service is to adapt the output of the system to the user's experience. This is achieved by building both static and dynamic user profiles. The static user profiles contain a number of parameters that can be manually set. Unlike the static profiles, the dynamic ones contain a series of parameters whose values are learnt automatically. The system registers a series of actions the users or carers perform with the text. For example, they can accept or reject the decisions performed by other software components. Based on editing operations a dynamic user profile will be built incrementally by the system. Because at this stage of the project the details of the dynamic profile are not yet fully specified we focus on the static profile in this section.

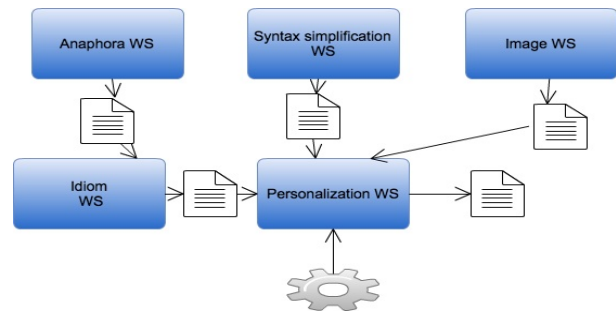The architecture of the Personalization component is presented in Figure 5.

Figure 5: The Personalization Web Service.

In addition to the web services presented in the previous sections (The Idiom Web Service and The Image Web Service) the Personalization Web Service receives input from Anaphora Web Service and Syntax Simplification Web Service. The Anaphora component resolves the pronominal anaphora and the Syntax Simplification component identifies and eliminates difficult syntactic constructions. The Personalization component aggregates the input from all web services and based on the parameters specified in the static profile (the wheel in Figure 5) transforms the aggregate document according to the user preferences. The personalization parameters in the static profile are the following:

1. *Image Disambiguation Confidence*. The image annotation is dropped when the corresponding concept disambiguation confidence is less than the threshold.

2. *Image Retrieval Confidence*. The image annotation is dropped when the assigned image is retrieved with a confidence lower than the threshold.

3. *Idiom Confidence*. The idiom annotation is dropped when the assigned idiom confidence is less than the threshold.

4. *Anaphora Confidence*. The pronominal anaphora annotations are dropped when the anaphor is solved with a confidence less than the threshold.

5. *Anaphora Complexity*. The parameter assess the complexity of anaphors. If the anaphora

17

complexity score is less than the specified threshold it drops the resolved pronominal anaphora.

6. *Syntactic Complexity*. It drops all annotations for which the syntactic complexity is less than the threshold.

The user can also reject the entire output of a certain web service if he does not need the functionality. For example, the user can require to display or not the images, to resolve or not the anaphora, to simplify the sentences or not, etc. In case the output of a certain web service is desired the user can specify the minimum level of confidence accepted. Any annotation that has a level of confidence lower than the specified threshold will be dropped. In addition to the parameters related to document content the static profile includes parameters related to graphical appearance (e.g. fonts or user themes) that are not discussed here.

5 Conclusions and further work

In this paper we presented three semantic components to aid ASD people to understand the texts. The Image Component finds, disambiguates and assigns Images to difficult terms in the text or related to the text. It works in two modes: automated or on-demand. In the automated mode a document is automatically enriched with images. In the on-demand mode the user highlights the concepts (s)he considers difficult and the web service retrieves the corresponding images. Further development of this component will involve disambiguation against Wikipedia and retrieval of images from the corresponding articles. The Idiom Component finds idioms and other figurative language expressions in the user documents and provides definitions for them. Further versions of the component will go beyond simple matching and will identify other categories of figurative language. The Topic Models component helps organizing the repository collection by computing topics for the user documents. Moreover it also offers a summarization of the document before the actual reading experience. Finally the Personalization component adapts the system output to the user experience. Future versions of the component will define dynamic user profiles in addition to the static user profiles in the current version.

Our hope is that the Open Book tool will be useful for other parts of populations that have difficulties with syntactic constructions or semantic processing, too.

References

Tony Attwood. 2007. *The complete guide to Asperger Syndrome*. Jessica Kingsley Press.

Simon Baron-Cohen. 2001. Theory of mind and autism: a review. *Int Rev Ment Retard*, 23:169–184.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. 2011. *Text Processing with GATE (Version 6)*.

Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, June.

K.H. Douglas, K.M. Ayres, J. Langone, and V.B. Bramlett. 2011. The effectiveness of electronic text and pictorial graphic organizers to improve comprehension related to functional skills. *Journal of Special Education Technology*, 26(1):43–57.

Brenda Fossett and Pat Mirenda. 2006. Sight word reading in children with developmental disabilities: A comparison of paired associate and picture-to-text matching instruction. *Research in Developmental Disabilities*, 27(4):411–429.

William Matthew Gillispie. 2008. *Semantic Processing in Children with Reading Comprehension Deficits*. Ph.D. thesis, University of Kansas.

Temple Grandin. 1996. *Thinking In Pictures: and Other Reports from My Life with Autism*. Vintage, October.

Temple Grandin. 2009. How does visual thinking work in the mind of a person with autism? a personal account. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1522):1437–1442, May.

Rajesh K. Kana, Timothy A. Keller, Vladimir L. Cherkassky, Nancy J. Minshew, and Marcel Adam Just. 2006. Sentence comprehension in autism: Thinking in pictures with decreased functional connectivity.

B. Lopez and S. R. Leekam. 2003. Do children with autism fail to process information in context ? *Journal of child psychology and psychiatry.*, 44(2):285–300, February.

Geoffrey Nunberg, Ivan Sag, and Thomas Wasow. 1994. Idioms. *Language*.