Introduction to Scientific Writing Assistant (SWAN) – Tool for Evaluating the Quality of Scientific Manuscripts

Teemu Turunen

Master's Thesis

ITÄ-SUOMEN YLIOPISTO

School of Computing

Computer Science

May 2013

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Joensuu
School of Computing
Computer Science

Turunen, Teemu Petteri: Introduction to Scientific Writing Assistant (SWAN) – Tool for Evaluating the Quality of Scientific Manuscripts
Master's Thesis, 102 p., 4 appendices (18 p.)
Supervisors of the Master's Thesis: PhD Tomi Kinnunen
May 2013

Abstract: Scientific Writing Assistant (SWAN) is a tool for evaluating and improving the quality of scientific manuscripts in English. SWAN is a rule-based, computer-assisted tool that employs sets of text quality metrics, and natural language processing tools. Other text quality metrics, and both manual and automatic tools that use them, exist, but mainly for grading essays. Also, unlike some other tools, SWAN does not give overall grade for a paper, but instead focuses on the local level feedback for the parts of a scientific paper that create first impressions: the title, abstract, introduction, conclusions, and the structure (headings and subheadings) of a paper. SWAN also evaluates the fluidity and cohesion of a text. The aim of this thesis is to provide a detailed, technical view for the metrics, as well as the tool itself. To evaluate the user experience of this tool, we conducted a study with the users of SWAN. According to the results of this study, SWAN is experienced useful as a tool for improving the quality of scientific writing.

Keywords: scientific writing tool, text quality metric, evaluation, natural language processing, fluidity

CR Categories (ACM Computing Classification System, 1998 version):
A.1, I.7.1, I.7.2, I.2.7

Preface

This Master's thesis was made for the School of Computing, University of Eastern Finland in Spring 2013.

Writing this Master's thesis was a suggestion from PhD Tomi Kinnunen to me a year ago. As I had been more or less involved with the topic of this thesis, Scientific Writing Assistant (SWAN), for the last few years, I thought I would possess sufficient amount of background knowledge to make the process easier, and therefore I agreed.

I was only partly right. The experience from developing SWAN for the past few years certainly helped. However, there was, and is, an incredible amount things to learn about scientific writing and reading.

The writing process had its ups and downs. I failed to schedule the process so that the workload would distribute evenly: I started too slowly, and at times, overestimated my capabilities. As I had my full time job, I only had few hours at the evenings during the workdays, and the weekends to spend for this. In the end I had to speed up, and use almost all of my waking time with this thesis, to get it done in time. It was a tiring, but a rewarding, and I am sure, a useful experience.

The SWAN project as a whole was a unique experience, of which I would like to thank everyone involved. I would also like to thank the lecturers and other university staff who taught and helped me along my way.

For the extremely valuable feedback and help with this thesis, I would like to thank my supervisor, PhD Tomi Kinnunen. I would also like to thank PhD Jarkko Suhonen for providing feedback for the study section of this thesis, and Mr. Jean-Luc Lebrun (in addition to his help during the development of SWAN) for allowing us to use his course's participants in our study.

Contents

1 Introduction

Scientific writing is hard. Yet, it is an essential activity for communicating science, sharing research findings, and making information available to other scholars (Elserag, 2006). A quote from Gopen (2004) summarizes well why it is difficult to produce a piece of scientific discourse:

> "The perfect piece of literature, when read by 1000 readers, should result in at least 1000 interpretations. The perfect piece of writing in the professional world, when read by 1000 readers, should produce one and only one interpretation."

Readers form interpretations when they read a written discourse (Gopen, 2004; Lebrun, 2011). The interpretations do not necessarily match those intended by the writer. When this happens, the reader may misunderstand the point the writer is making. And when this happens, the discourse does not effectively do what it was meant to: transfer the thoughts and rationale of the writer to the reader. Thus, the writer should write so that the possibilities to misinterpretations decrease. One way to achieve this is to increase the readability of the discourse.

Defining and measuring text quality and readability has a long history and tradition (Pitler and Nenkova, 2008). There has long been consultation and training available for clear writing, e.g. Robert Gunning Associates started offering 1944 consultation for newspapers, magazines and corporations (Pitler and Nenkova, 2008). During the years linguists have studied text flow, cohesion building devices in English language, and formed theories such as *rhetorical structure theory*, and *centering theory* (Pitler and Nenkova, 2008). After observing how great impact vocabulary in text has on its readability, different metrics have been developed (Pitler and Nenkova, 2008).

The early works measured text readability with predefined lists of the most frequent words in language: the more frequently occurring words a given text contained, the more readable it was determined (Pitler and Nenkova, 2008). Since the most frequently occurring words are often short, metrics and tests was developed, in which readability was formally linked with length of words (Pitler and Nenkova, 2008). These include, for instance, *Flesch-Kincaid* readability test, *Automated Readability Index* (ARI), *Simple Measure of Gobbledygook* (SMOG), *Gunning Fog*, and *Coleman-Liau* indices

(Pitler and Nenkova, 2008). The reliability and accuracy of the simplest readability tests have been questioned in studies (Feng et al., 2010; Pitler and Nenkova, 2008). Later on, more accurate and complex language models, including trained classifiers such as support vector machines (SVMs), have been developed (Pitler and Nenkova, 2008).

Despite the history, and various readability tests, indices, and models, no unified computation models that consider multiple aspects of readability, exist (Pitler and Nenkova, 2008). Instead, most studies have focused on models for single factor affecting readability, and for specific audience (Pitler and Nenkova, 2008). Study by Pitler and Nenkova (2008) concentrated on analyzing vocabulary, syntax, cohesion, entity coherence, and discourse. The relations in discourse were found to be an strong indicator of readability. Other factors that correlated with perceived quality, were the average number of verb phrases in sentences, the number of words in a paper, and the amount of occurrences of domain-specific words (Pitler and Nenkova, 2008).

To automate the readability metrics, computerized tools have been developed. Tools' purposes and intended audiences vary from essay grading for high school students to assessing scientific papers for academics. For instance, *Criterion Online Writing Service* represents the former (Burstein et al., 2004), while *AMiable Article Development for User Support* (AMADEUS) represents the latter (Aluísio et al., 2001). These tools, in general, employ text quality metrics, natural language processing tools for detecting discourse elements, and statistical models for e.g. calculating probabilities of occurrences for certain discourse elements. According to our literature review, not much focus has been given for developing text quality metrics and tools specifically for scientific papers. This thesis, however, describes one such tool.

Scientific Writing Assistant (SWAN) is a rule-based, computer-assisted tool that combines text quality metrics and natural language processing. SWAN provides feedback on the parts of a scientific paper that create the first impressions: the title, abstract, introduction, conclusions, and the structure (headings and subheadings). These metrics are designed by Lebrun (2011). SWAN does not give overall grading for a paper. Instead, SWAN points out problems at the local level, as well as assesses text fluidity (both automatic and manual options are available) and cohesion. The newest SWAN versions also contain metrics for assessing the relationships between visuals (figures and tables) in a paper. SWAN works only on texts written in English, the language that is used in scientific writing by the vast majority. In natural language processing SWAN

relies upon Stanford NLP tools, Parser (Klein and Manning, 2003) and POS Tagger (Toutanova et al., 2003).

The aim of this thesis is to introduce SWAN by offering a detailed technical view: while the book by Lebrun (2011) describes the metrics, and offers extensive reasoning for how and why they improve text quality, it does not contain many technical details. Kinnunen et al. (2012) has also written a paper about SWAN for the Conference of the European Chapter of the Association for Computational Linguistics, but the paper, too, does not delve into technical details. Thus, this thesis is an attempt to bridge the gap between the already published literature, and the current state of the implementation of the SWAN project.

We conducted a study with the users of SWAN. The aim of the study was to find out how well SWAN performs as a scientific writing tool, and what kind of problems the users have with SWAN. With these answers we can have more pointers to the parts we should focus on more to make SWAN yet more useful.

The thesis is organized as follows. In Section 2 we will discuss shortly about the background, problems, and related solutions for assessing text quality. In Section 3 we will have a detailed look on the text quality metrics that are implemented in Scientific Writing Assistant. We will go through the metrics related to the standard parts of a scientific paper: the title, abstract, introduction, and conclusions. In addition to this, we will discuss metrics for the structure of a paper, as well as metrics for measuring the fluidity of a text. SWAN, as its latest feature, also contains a set of metrics to assess the visuals (figures and tables) in a paper; however, as these are relatively new additions with only a few user experiences, these will not be discussed in this thesis. After the metrics section, in Section 4, we will discuss about the current implementation of SWAN. We will have a short look on the basic use flows, and other implementation specific matter. We will also discuss about the natural language processing tools, SWAN employs. The last section before the conclusions and future work, Section 5, presents the study we conducted on the tool. In the end, we will conclude this thesis, and suggest some improvements as a future work for the SWAN project.

2 Scientific Writing and Reading

Compared to the general written communication humankind has engaged in, the scientific writing is relatively new activity: the first scientific journals appeared not until 1665, and the standard paper organization IMRaD (Introduction, Methods, Results, and Discussion) was developed within the past 100 years (Day, 1998).

Scientific writing requires various skills from the scholars, namely ability to accurately communicate ideas, procedures, and findings, and ability to relate and interlink evidence (Shah et al., 2009). Writing is essential for sharing research findings, and making information available to others (Elserag, 2006). Publishing academic papers is also a measure of productivity that can be used, for instance, when assessing for promotion (Elserag, 2006). In addition to these external factors, the scientist itself benefits from engaging him/herself into writing activities: many think that only way to become experts in their field, is to involve themselves deeply in the literature, and contribute to it (Elserag, 2006).

2.1 Problems and Difficulties

Scientific writing is a demanding activity. Several studies have aimed in identifying the problems related to writing. The main problems revealed by a study by Shah et al. (2009), where students performed writing activities, were 1) problems related to the structure of discourse, and 2) cognitive burden caused by writing activities. Also other studies (Elserag, 2006; Aluísio et al., 2001; Pitler and Nenkova, 2008) have found structure-related problems. These problems relate to difficulties in distinguishing between the content and the structure (Shah et al., 2009), using rhetorical structures from the writer's mother language in English written discourse (Aluísio et al., 2001), and difficulties in comprehending complex syntactic structures (Pitler and Nenkova, 2008). Also, Lebrun (2011) and Gopen (2004) have noted the impact of structure complexity in reading comprehension. Findings by Lebrun (2011) include undefined acronyms, synonyms, and structural elements belonging together separated by too many words (e.g. verb and its object, subject and its verb, pronoun and its noun). Gopen (2004) uses the term "reader energy" to describe mental resources that are reserved and used while reading. These resources are used to both comprehend the structure of discourse, and its meaning; the more is required to decipher the other, the less there is left for the

other (for more details, see Section 3.6). Rare words, and technical terms have also been found to affect reading comprehension (Pitler and Nenkova, 2008; Lebrun, 2011).

Problems caused by cognitive burden originate from both subjective (within individual), and objective (associated with the writing task) constraints (Shah et al., 2009). Some of the subjective constraints have been identified in studies by Elserag (2006); Shah et al. (2009); Pololi et al. (2004); Witt et al. (1995). While the list below was mainly a result of studies regarding students (novice writers), it also concerns, for certain amount, those with more experience:

- Perfectionism
- Lack of general writing experience
- Lack of academic writing experience
- Lack of confidence in one's abilities
- Anxieties originated from writing activities
- Sensitivity or resistance to feedback
- Perceiving writing skills useless after graduating
- Bad writing experiences, that make writing unpleasant
- Fear of failure

Some objective and external constraints have been identified both among students, and professionals. Studies by Sprague et al. (2003) and Rodgers and Rodgers (1999) found that time constraints, ongoing status of studies for students, issues of co-authorship, institutional policies, and work pressure are among the major reasons for failures in writing activities. Elserag (2006) also lists teaching responsibilities, committee assignments, personnel disputes and grant deadlines reasons for difficulties in writing for professional researchers. Also, acting between the rather strict boundaries of scientific research and writing add to the challenges perceived by the writer: one must attend to the soundness of subject matter, keep audience in mind, and, at the same time, take care of the clarity, style, structure, and precision of the written scientific discourse (Shah et al., 2009).

Other reasons for difficulties are related to teaching scientific writing, and to the school system (Chuck and Young, 2004). Students are confused by differences between styles of teaching science and scientific writing between subjects and/or instructors (Chuck and Young, 2004). Chuck and Young (2004) also found out that having the students

use real scientific papers as models, in some situations confused students even further: they could not differentiate different writing styles and structures from the papers. Due to these problems, students' writing skills remain underdeveloped, which results in further problems in later grades. Chuck and Young (2004) also point out that students tend to focus on achieving passing grades on writing assignments, instead of considering writing as a way of improving their skills, and understanding of the subject.

2.2 Solutions

Tools and methods meant for easing the challenges related to reading and writing, and for improving the quality of papers, can be divided into two general categories: manual solutions, and computerized semi-automatic to automatic tools. Manual solutions include using books, and other user-friendly documentation, training and mentoring, using writing strategies, working on groups, and peer-reviewing.

Books, such as "Scientific Writing A Reader and Writer's Guide" by Lebrun (2011), "Scientists must write" by Barrass (2002), "Guide to publishing a scientific paper" by Körner (2008), "How to write and publish a scientific paper" by Day (1998), and "The Craft of Scientific Writing" by Alley (1996), to name a few, focus on giving advice directly on scientific writing. Other books, such as "Expectations: Teaching Writing from a Reader's Perspective" by Gopen (2004), and "The Elements of Style" by Strunk Jr (1918) offer advice on general writing, but they can be applied, to a certain extent, to scientific writing as well. See the references section of this thesis for more examples.

Training and mentoring have been shown to effectively help students (Elserag, 2006; Shah et al., 2009); especially when teaching domain-specific reading strategies (Elserag, 2006). Both the role of a mentor and the feedback received are essential (Shah et al., 2009). Chuck and Young (2004) developed a cohort driven assessment tool for university students. In their tool, a class of students prepare a paper, to which they receive feedback from the instructor. After submitting their papers, the instructor goes through them, and based on the writing problems found from them, develops a working scheme specifically for the class group. This scheme is given to the students for reworking their submitted paper. With this scheme, and a mixture of peer-review and self-review, the students then resubmit their paper. Chuck and Young (2004) found

out that this methodology was effective in improving the readability of the papers.

Even simple writing strategies, such as ignoring structure, and grammar when writing the first draft, can help (Elserag, 2006). Shah et al. (2009) also found out that *backward design* of manuscripts can release some of the writing blocks. This means, that the writer tries to visualize the writing project from an overall perspective, and see the goal. After this, the writer plans, and develops steps to fill the gap between the goal and the current situation.

Another finding from the study by Shah et al. (2009) was that working in groups eases the difficulties experienced by students. Group working has been experienced as encouraging and motivating. According to Chuck and Young (2004), student-centered learning accomplishes greater student engagement. Another study cited in Shah et al. (2009) also indicated that pair working results shorter, but more complex, accurate, and higher-quality texts. Shah et al. (2009) also describes further studies that show that peer support groups are not useful only for the students, but also faculties' publication frequency has been shown to increase by emphasizing group work, and collaboration.

Peer-reviewing is the standard step in academic publication process. Its objectives are to prevent publication of bad work, improve scholarship, language, and data presentation (Szklo, 2006). As it is, it both prevents less quality papers from being published, and increases paper quality in a form of peer-reviewer's feedback (Szklo, 2006). However, Szklo (2006) argues that the assessment of quality in peer-reviewing is elusive: although many journals have quality items (such as originality, design, importance, and presentation), they do not instruct reviewers *how* to use them. The peer-reviewing's reliability, and validity also remain undetermined (Szklo, 2006).

In addition to manual tools, excessive amount of computer-assisted tools have been developed. Their purposes and intended audience vary. Most of the tools that were brought up in literature relate to improving the quality of essays for students of various grades. These are for example *Criterion Online Writing Service* (Burstein et al., 2004), *MarkIT*, *Project Essay Grade* (PEG), and *Intelligent Essay Assessor* (IEA) (Williams and Dreher, 2005). Some tools, namely works of Si and Callan (2001) and Collins-Thompson and Callan (2004), are also developed for capturing and assessing scientific texts from web pages (Feng et al., 2010). There are not that many tools that directly focus on scientific writing and papers. One such tool, in addition to our Scientific Writing Assistant, is the *AMiable Article Development for User Support* (AMADEUS)

(Aluísio et al., 2001).

These tools employ various methods and technologies. Critique writing analysis tools, which are part of Criterion Online Writing Service, uses corpus-based, and statistical methods (Burstein et al., 2004). The *e-rater*, also a part of Criterion Online Writing Service, uses a combination of natural language processing, and statistical tools (Williams and Dreher, 2005). MarkIT also employs NLP tools, and an electronic thesaurus. Intelligent Essay Assessor is based on latent semantic analysis (LSA), and works on the vocabulary of texts (Williams and Dreher, 2005). Latent semantic analysis can be used to examine similarity between passages of a given text. It is a corpus-based statistical method, which focuses on conceptual content, rather than surface features such as word frequencies (Kakkonen and Sutinen, 2004). In addition to IEA, it is used in the study of Kakkonen and Sutinen (2004) for essay grading. AMADEUS consists of three tools: Critiquing tool, Reference tool, and Support tool (Aluísio et al., 2001). These tool parse linguistic features from the text by using similarity metrics.

3 Formalization of quality metrics for scientific manuscript evaluation

Scientific Writing Assistant (SWAN) uses formalized text quality metrics designed by Lebrun (2011). These metrics are tested on 960 scientists since 1997 (Kinnunen et al., 2012). Currently, there are metrics dedicated for the standard parts of a scientific paper: the title, abstract, introduction, and conclusions. In addition, Lebrun (2011) has designed metrics that assess the structure (outline) of a paper, the visuals in a paper, as well as the fluidity of a given text. These metrics, excluding the visuals metrics, are described in the following sections in a form of pseudocode.

The pseudocode is, when suitable, abstracted and/or simplified to benefit the reader the most. This means, that for some metrics, there is only a verbal, high-level explanation, while for the others, there is a more formal and accurate description. For the most part, the pseudocode consists of `IF`, `THEN`, and `WHERE` statements. The `IF` statements describe a condition, which must the fulfilled, and the `THEN` statements the result for fulfilling the condition. These results are, for the most part, boolean flags, that may be used as an input for other metrics, or for showing messages to the user. The `WHERE` statements describe implementation specific details, such as constant values. Some metrics contain a line starting with `MANUAL`: these metrics require manual work from the user. A simplified example of these metrics is given in Listing 1.

```
IF
  title contains attractive words AW
THEN
  TITLE_ATTRACTIVE = TRUE;
WHERE (in the current implementation): AW=<a list of words
   indicating attractiveness>;
```

Listing 1: An example of a pseudocode for a metric.

3.1 Title metrics

A title is the shortest part of a scientific paper; yet it has many important purposes and roles. A title is a tool for search, it states contribution, helps to form the first impressions of how well paper could satisfy needs, and to assess the knowledge level

needed to benefit from reading the paper, and reveals the kind (genre, breadth and depth) of the paper (Lebrun, 2011).

For a reader, the title helps to make the decision of dropping or reading on the paper (Lebrun, 2011). For this purpose, the title states and gives first idea of the contribution and provides clues on purpose, specificity, scope, impact and overall nature of the paper (Körner, 2008; Lebrun, 2011). Using this information the reader can then assess whether they can benefit from the paper.

For a writer, the title is the place where they can add search keywords in order to make the title as searchable as possible. A title can also act as an attention catcher and attract targeted readers and filter out those not targeted. A title is also used to differentiate a paper from the others (Lebrun, 2011).

In order for a title to best fulfill these roles, a title should be "unique, lasting, concise, clear, honest, representative, catchy and easy to find" (Lebrun, 2011). For the most part, these qualities will be covered in the following sections with accompanying metrics that make the quality measurable.

Understandability

Title understandability directly relates to how well and easily a reader can comprehend the subject and purpose of a paper. Since the title plays a vital role in the decision making process a reader goes through when assessing the benefits of reading the full paper, whether the title is understandable enough or not can make a difference between the decision to read further and reject the paper.

A clear title reveals the genre, breadth and depth of a paper without straining the cognitive abilities of the reader. This also allows the reader to form a relatively accurate picture of the paper. An unclear title can contain the same properties, but causes the reader struggle in understanding them and may lead in misinterpretations as the reader has to start guessing in place of deducting based on the given hints. In the worst case the reader misinterpretates the whole purpose of the paper badly and may get a negative picture of the whole paper regardless of how well the paper in reality presents the research and its results. But then, when is a title easily understandable? Human intuition, especially a trained one, can be moderately accurate, but is also subjective

and therefore not suitable for generalizations. Fortunately, there are also means of excluding some of the subjectivity out of the formula: instead of relying on pure intuition, title understandability can be made measurable by examining the contents of a title. Lebrun (2011) has developed metrics for this very purpose. These metrics are described in Listings 2 and 3.

The first of them, the metric in Listing 2, measures title *clearness*. First of all, the metric calculates the length of the title in characters and in words. The longer the title, the more time it will take to read it and the more unclear it may become. If the title is considered long, it also counts prepositions and punctuation marks. Punctuation may be used to divide otherwise long expressions, and prepositions to clarify long modified nouns. Having low amount of punctuation and prepositions in a long title may indicate low clearness for the title. Ambiguous prepositions such as “and” and “with” and long noun-phrases without attractive verbal forms may also make title feel tedious and increase the risk of misinterpretation.

```
IF
  title character count > TC, OR
  title word count > TW AND percentage of prepositions and
     punctuation marks in title < P%, OR
  title contains ambiguous prepositions AP, OR
  longest noun-phrase without preposition or attractive verbal
     form has over NPW words
THEN
  TITLE_UNCLEAR = TRUE;
WHERE (in the current implementation): TC=100, TW=6, P=25,
   AP={"and", "with"}, NPW=3;
```

Listing 2: Calculating title clearness

The other metric (in Listing 3) also relates closely to clearness by extending the title length consideration by taking a closer look at the *conciseness*. Since a title can only consist of a limited amount of words and should state contribution as clearly as possible (Lebrun, 2011), expressing the necessary in a concise manner is vital. Conciseness can also reduce the cognitive burden of the reader by decreasing the amount of words they must store into their working memory (Gopen and Swan, 1990; Daneman and Carpenter, 1980; Daneman and Merikle, 1996). This metric concentrates on looking for unnecessarily verbose expressions, such as “study of”, that lengthen the title without bringing much informative value. In these cases, title clearness may be improved

by removing the expressions or, in case of lack of prepositions and punctuations, by adding these word classes to bring clarification to long phrases.

```
IF
  title contains overlong non-concise expressions NCE
THEN
  TITLE_HAS_NONCONCISE_EXPRESSIONS = TRUE;
ELSEIF
  title contains overlong non-concise expressions NCE, AND
  TITLE_UNCLEAR == TRUE AND percentage of prepositions and
     punctuation marks in title < P%:
THEN
  TITLE_NOT_CONCISE_AND_NOT_CLEAR = TRUE;
WHERE (in the current implementation): NCE={words "a", "an",
   "study of", ignoring the word case}; P=25;
```

Listing 3: Determining title conciseness

Searchability

One of the purposes for a title is to help potential readers to find the paper it belongs to; the parts responsible for this are called *search keywords*. Simply put, a search keyword is a certain word in title that has some informative value and can therefore be used in searches. These keywords are used by readers when they do queries with their favorite scientific paper search engines. Obviously then, much depends on how well the title covers possible keyword combinations readers use when searching papers. Also, an important factor for search success is what kind of search keywords are included in the title. Lebrun (2011) mentions three kinds of keywords: general, intermediary and specific (Figure 1).

General keywords, as the name suggests, are basic terms that are used to describe a certain domain. Since they require only a basic level of domain knowledge, they have the potential to gain a larger audience. Therefore, as search keywords, they are also more frequently used. However, this frequency may also bring problems: general keywords by themselves may not be enough to differentiate the title from others titles, therefore making finding the title difficult. *Intermediary keywords*, on the other hand, require deeper understanding of the domain they are used in, and therefore appear less frequently in titles. They are often associated with methods that are used in multi-

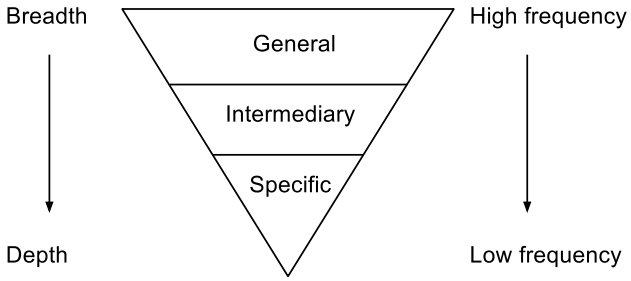

Figure 1: Keyword categories: general, intermediary, and specific. The general keywords are basic terms, with a large breadth. The more specific keywords goes, the more it has depth. At the same time, the frequency of keyword occurrence decreases. (Lebrun, 2011)

ple research fields. The last category contains the *specific keywords*, which are used by experts to describe terms that require deep understanding of the domain. Specific keywords are best for differentiating titles, but they also require more background and domain knowledge therefore possibly making title hard to find for non-expert readers. However, the division between categories is not constant: which keyword belongs to which keyword category depends on domain, and even journal where the paper is (to be) published. Also, the background knowledge of the reader affects how a reader experiences each keyword.

But how, then, should a title be constructed? A metric by Lebrun (2011) described in Listing 4 gives one possible answer to this question. A title, first of all, should contain more than one search keyword: without or with only one search keyword title will be hard, if not impossible, to find. Second, as mentioned before, neither general, intermediary nor specific keywords may be enough by themselves. Instead, a title should have at least two kinds of keywords (Lebrun, 2011): more generic keywords to attract non-expert readers, and more specific keywords to differentiate and attract expert readers. Third, a title should not have too many specific keywords with respect to other categories: a title with too many specific keywords makes it hard to find and hard to understand for non-experts because they do not necessarily have enough background knowledge for these keywords. This relates to title clearness as well: a title should be clear to both non-experts and experts.

```
MANUAL: User defines keywords and sets them search categories
   {none, generic, intermediary, specific}

IF
```

```
  search keyword count < N, OR
  search keyword count == M AND specific search keyword count ==
     0
THEN
  TITLE_HARD_TO_FIND = TRUE;
WHERE (in the current implementation): N=1, M=2

IF
  generic search keyword count > 0 AND intermediary search
     keyword count  > 0 AND specific keyword count > 0
THEN
  TITLE_KEYWORDS_WELLSPREAD = TRUE;

IF
  specific keyword percentage > P%
THEN
  TITLE_HARD_TO_FIND_FOR_NONEXPERTS = TRUE;
WHERE (in the current implementation): P=70;
```

Listing 4: Search keywords

Attractiveness

Lebrun (2011), in his book, uses face as a metaphor to describe the role and properties of a title. They do have something in common: a title is most of the time the very first part of a paper one sees and therefore acts as the first interface between a reader and the paper. In the real world, a face gives the viewer the first impressions and sets expectations about the person. A title does the same for the paper. These first impressions are vital when a reader or a reviewer forms an opinion about the paper and ultimately makes a decision whether to read further. The more attractive a title, the better probability it has to be noticed among plethora of other titles.

But how can this seemingly subjective term “attractiveness” be measured? Lebrun (2011) has formed a metric to qualify attractiveness in a title; this metric is described in Listing 5. This metric calculates word classes (such as numerals, adjectives and adverbs), and verbal forms (gerundives and infinite forms), which increase attractiveness, from the title to form an attractiveness level. User-defined keyword search categories are also taken into account: non-search keywords can be attractive because they usually

belong to another field or domain.

```
MANUAL: User defines keywords and sets them search categories
    {none, generic, intermediary, expert}

IF
  title has numbers OR adjectives OR adverbs, OR
  title has gerundives OR infinite verbs, OR
  title has keywords not used for search
THEN
  TITLE_ATTRACTIVE = TRUE;
ELSE
  TITLE_ATTRACTIVE = FALSE;
```

Listing 5: Title attractiveness

Contribution and other sections

A paper, whether for a conference or a journal (or another occasion), is written for a reason. This reason usually contains a desire to announce and get readers to read the results of an conducted effort considered as contribution: a new method, algorithm, application, approach, theory or a finding of another sort; the sort of the contribution does not matter, as long as it provides something new to the field. To get as many readers as possible to explore the fruits of an effort, the scientist then writes a paper. With the help of search keywords readers will find the paper. When they read the title, they want to get the first idea of what the paper is about. And what they expect to see first, is the reason for the writer to write and for them to read the paper. In other words, they expect to see the contribution (Lebrun, 2011). Therefore, the contribution should be the first thing mentioned in the title.

There are, however, some exceptions. The recommendation for upfront placement for contribution concerns mainly titles that are incomplete sentences (those lacking a conjugated verb). The minority of titles (e.g. those used in life sciences), however, are full sentences. In these cases, the contribution should start at the first verb and continue until the end of the sentence (Lebrun, 2011). Listing 6 describes metrics, also by Lebrun (2011), that make automatic checks for title to see if their contribution is placed correctly.

```
MANUAL: User defines keywords and sets which represent
    contribution
IF
  title is NOT a sentence
THEN
  IF
    title has keywords defined AND first title keyword
        represents contribution
  THEN
    CONTRIBUTION_AT_CORRECT_PLACE = TRUE;
  ELSEIF
    title has keywords defined AND first title keyword DOES NOT
        represent contribution
  THEN
    CONTRIBUTION_AT_CORRECT_PLACE = FALSE;
ELSEIF
  title is a sentence
THEN
  IF
    keyword representing contribution comes before verb
  THEN
    CONTRIBUTION_AT_CORRECT_PLACE = FALSE;
```

Listing 6: Contribution placement

Another issue in addition to contribution placement is how the keywords representing contribution are placed relative to each other. The metric handling this issue is described in Listing 7. The ideal would be that the contributive keywords be next to each other, in one group. A scattered contribution can be an indicator for multiple contributions in one paper; in these cases it may be better to write multiple papers to address these contributions separately (Lebrun, 2011).

```
MANUAL: User defines keywords and sets which represent
    contribution

Search title keywords for the first, and last occurrence of
    contributive keywords, and store indices of those keywords to
    INDEX_FIRST and INDEX_LAST. Create a list TKW, which contains
    title keywords from range [INDEX_FIRST, INDEX_LAST].

IF
  TKW contains keyword that DOES NOT represent contribution
```

```
THEN
  CONTRIBUTION_SCATTERED = TRUE;
```

Listing 7: Contribution scatterance

A title can, and usually should, also contain other sections than the one containing the information about the contribution (Lebrun, 2011). These sections (as seen in the Listing 8 that contains pseudocode for the metric to check these sections) may be used to provide additional hints about the research and its impact, main application, used methodology and results. These sections increase the informative value of a title and thus make it more useful and attractive to a reader. If a section is not present in the title, the user is informed, and asked, whether they should be in the title.

```
MANUAL: User identifies and defines parts that should be found
    in a title: "impact of research", "main application of
    research", "used methodology" and "results or section
    corresponding to contribution"

Check user selections.

IF
  a part is missing
THEN
  COMPLETE_TITLE = FALSE;
```

Listing 8: Other title sections

3.2 Abstract metrics

An abstract is, in most cases, the second part of a paper, located after the title. While an abstract may have more words than a title (around 200 to 250 words, depending on journal or occasion), the amount is still limited and require concise writing style (Kurmis, 2003; Körner, 2008). As many readers only read abstracts, and many reviewers gain their first impressions from an abstract, it should provide an overview of all the prominent elements of the paper (Kurmis, 2003); for the same reason, an abstract should also stand alone (Körner, 2008; Lebrun, 2011). An abstract has also a function in searches as many search engines and scientific article databases provide the abstract with the title and bibliographical data.

In addition to the purposes mentioned in the previous paragraph, abstract can also be used to clarify the title, to provide details on contribution, to help reader make decision of rejecting or reading further on the paper, and to guide the life of paper, when written early (Lebrun, 2011). In order to meet these purposes, an abstract should contain the following qualities (Lebrun, 2011): it should be tied to title (i.e. expand and repeat the key points of a title), should be complete (the four sections; we will discuss these shortly), concise (for the word limit), stand-alone, be representative for the whole paper (the first impressions), precise to convince reader of the benefits of the paper, and present for allowing the abstract remain attractive. The following sections discuss these qualities in more detail and provide formal metrics for making these qualities measurable.

Consistency between Title and Abstract

The title and the abstract are the first two parts of a regular scientific paper. As such both have a common aim in attracting a reader to read further as well as introducing the reason for writing the paper concisely. The title makes the first announcements, but with very limited amount of words to use. This limitation makes titles faster to read, but does not allow clarification for difficult terms. Thus, the abstract with larger capacity can and should provide further definitions (Lebrun, 2011). For this reason, title and abstract should be consistent with each other, in other words, there should exist a continuum between them. The easiest way to increase consistency is to repeat and expand title keywords, i.e. the most important words of the title, in abstract.

The consistency between title and abstract can be measured with the metric by Lebrun (2011) described in Listing 9. The first part of the metric compares the title keywords and the first sentence of an abstract to define the coherence between the two. The comparison ignores word cases, and for maximum match-ability, uses word *stems* (root forms of a word). After comparisons, a percentage of title keywords in the first abstract sentence is calculated. The ideal range for coherence would be from 30 % to 80 % (Lebrun, 2011). Coherence percentages under 20 % mean that only one fifth of title keywords were found from the first abstract sentence, thus making title and abstract inconsistent and failing to meet the reader expectations (Lebrun, 2011). The other end of the range, the percentages from 80 % to 100 %, on the other hand, may indicate that the title keywords were simply repeated in the first abstract sentence without bringing

any additional informative value. This consumes the limited capacity of an abstract, and should therefore be avoided.

The latter part of the metrics measures the strength of *cohesion*. It compares the title keywords with the whole abstract, having the same comparison criteria as the former part of the metric. As a result, all title keywords should be found from the abstract. Failing to meet this aim reduces the strength of cohesion between title and abstract and may make them feel disconnected. Possible reasons for a missing keywords may be: 1) the missing title keyword is not important enough to be explained; in this case the keyword may be removed from the title to increase both consistency between title and abstract and conciseness of the title 2) instead of the same keyword, a synonym was used in abstract; using synonyms may decrease the relevance score calculated by search engines, thus lowering the paper placement in the search results ranking. In addition to missing title keywords, also frequently occurring abstract keywords that are not found in the title, may lower the coherence.

```
checkConsistencyFirstSentence(title, abstractFirstSentence):
  Calculate the percentage P of title keywords in the first
     sentence of the abstract. Count keywords in the first
     sentence of the abstract A1KWC. Count title keywords TKWC.

  Comparison will ignore word case and uses word stems. Also,
     comparison uses only nouns and verbs from the abstract.

  IF
    P < MIN_P%
  THEN
    TITLE_ABSTRACT_COHERENT = FALSE;
  ELSEIF
    P > MAX_P%, AND
    A1KWC < N * TKWC
  THEN
    ABSTRACT_FIRST_SENTENCE_REPEATING = TRUE;
  WHERE (in the current implementation): MIN_P=20, MAX_P=90,
     N=1.2;

checkConsistencyFull(title, abstract):
  Compare title keywords with the full abstract. Calculate the
     most frequently occurring (at least N times) abstract
     keywords NOT in the title into FAWC.
```

```
Comparisons will ignore word case and uses word stems. Also,
    comparison uses only nouns and verbs from the abstract.

IF
  NOT all title keywords are found from the abstract
THEN
  TITLE_ABSTRACT_COHESION_STRENGTH_LOW = TRUE;

IF
  FAWC > 0
THEN
  TITLE_ABSTRACT_REFLECTION_LOW = TRUE;
WHERE (in the current implementation): N=2;
```

Listing 9: Coherence between title and abstract

Completeness

Whereas a title, with a very limited amount of words, introduces the contribution, topic and possibly the background to the research, the more verbose abstract has the capacity to extend this and provide additional information. This information plays an important role in answering to the questions a reader has in mind after they have read the title and started assessing whether they have need to read the rest of the paper. A recommended structure for an abstract contains the following parts (Day, 1998; Katz, 2009; Lebrun, 2011; Baker, 2012):

1. The topic and aim of the paper
2. The methodology used in determining the results of the research
3. The results of research
4. The impact of research

The first part, the *topic and aim* introduces the reader briefly to the topic, the research problem and its background as well as the aims the researchers had set for the research. The second part, the *methodology* contains description of the methods used to solve the problem and achieve the aim. The third part, the *results* describes the main results and how well the problem was solved. The fourth and final part, the *impact* justifies why

the research was conducted and paper written by stating the benefits of the research for the scientific community as well as to the reader. Together these four parts answer to the reader's questions and increase abstract's informative value and may increase the probability for the whole paper to be read. Vice versa, an abstract lacking one or more parts may be considered incomplete and therefore may cause the paper seem not worth of paying, downloading and reading.

However, in some cases the part describing background to the contribution (usually placed in the part one) may not be necessary to be included to the abstract (Lebrun, 2011). Such situations may arise when the rather limited amount of words allowed for an abstract is reached and one needs to free words for describing more prominent sections, such as the impact of research (Lebrun, 2011), the part that has an important role in convincing the readers. Having an overly verbose description of less prominent sections is also one of the main reasons for missing a part; the other reasons include the author considering the mention of results being enough for determining the impact of the research, author being unable to assess impact caused by atomization of research tasks or having too small a contribution to reserve space from the abstract (Lebrun, 2011). Also, a review paper or a short paper may not necessarily has to have all the four parts.

The corresponding two metrics for checking the abstract completeness are described in Listing 10. They both require some manual effort from the user at the beginning as he or she has to define the sentences which reflect the mentioned four abstract parts. After the manual effort, the rest of the metrics are computed automatically. First of them basically looks over the user selections to measure abstract completeness and notifies in case of a missing part. The second metric looks for unnecessary parts that occupy room from more prominent matters, such as the words to ensure coherence between title and abstract, or the description of the impact the research results have. Freeing room for these more prominent matters by removing the unnecessary parts may increase the usefulness of an abstract (Lebrun, 2011).

```
MANUAL: User defines which of the following sections are found
   in the abstract: "background to the contribution", "main
   objective of research", "used methodology", "results or
   section corresponding to contribution", "impact of research".

IF
```

```
  one or more sections apart from background to contribution are
     NOT marked
THEN
  ABSTRACT_INCOMPLETE = TRUE;

IF
  title and abstract are not coherent (see
     checkConsistencyFirstSentence) AND background to
     contribution IS marked, OR
  background to contribution IS marked AND impact of research IS
     NOT marked
THEN
  ABSTRACT_HAS_UNNECESSARY_PARTS = TRUE;
```

Listing 10: Measuring the abstract completeness and informativeness

Attractiveness

An attractive abstract encourages, engages and convinces readers to read further. According to Lebrun (2011) abstract attractiveness can be increased and ensured with two things: writing the abstract in dynamic verb tenses and by providing sufficient amount of precision or detailed descriptions to the matters expected to be in abstract such as the main accomplishments.

The former attractiveness factor, the *dynamic verb tenses* are, in other words, tenses and verbal forms that make the sentences feel vibrant, lively and therefore engage the reader. Such impact can be achieved with present and, in certain cases, perfect present verbal tenses. The past tenses, on the other hand, are considered unexciting and may cause the paper feel dated (Lebrun, 2011). As conclusions are usually written also in past tenses, having both parts in the same verb tense may make the conclusions feel like a plain repeat for an abstract. Some, for instance Day (1998), however, recommended using past tense in an abstract. He explains that one should use past tense when referring to one's own work, which is not yet presumed to be established knowledge; an abstract mostly contains one's own work, and thus it should be written in past tense.

A metric by Lebrun (2011) described in Listing 11 examines the verb tenses in an abstract. The metric relies upon natural language processing (NLP) tools, which in the current implementation are provided by the Stanford Tagger library (Toutanova et al.,

2003). The Section 4.3 addresses the topic in more detail.

The ideal cases are: 1) an abstract written with only present tense or 2) with present and perfect verbs with section explaining the background to the contribution. An abstract should also be written with only few (in the current implementation for this metric the amount is 2) different tenses: having too many different tenses may confuse the reader and make it feel unattractive.

```
MANUAL: User defines which of the following sections are found
    in the abstract: "background to the contribution", "main
    objective of research", "used methodology", "results or
    section corresponding to contribution", "impact of research".

Examine all verb tenses in abstract. Count each distinct tense.

IF
  only present verb tenses in abstract
THEN
  ABSTRACT_DYNAMIC = TRUE;
ELSEIF
  abstract written with present and perfect present verbs, AND
  background to contribution marked
THEN
  ABSTRACT_DYNAMIC = TRUE;
ELSEIF
  abstract written without present tense
THEN
  ABSTRACT_DYNAMIC = FALSE;

IF
  distinct verb tense count in abstract > N
THEN
  TENSES_VARIED_TOO_MUCH = TRUE;
WHERE (in the current implementation): N=2
```

Listing 11: Choice of verb tenses

The other factor impacting attractiveness is how precise and detailed the abstract is. A title must be concise and therefore may not have capacity for precision. The reader, however, expects precision and more detailed descriptions after reading the title, and therefore an abstract should provide them (Lebrun, 2011). The precision and detail

allows to meet these expectations and convinces reader of the benefits of reading the whole paper. A good way to bring precision to the text is to use numbers as they are objective and unambiguous (Lebrun, 2011; Katz, 2009). The metric in Listing 12 presents pseudocode for examining the precision (the numbers) of an abstract. When numbers cannot be used, e.g. in case of descriptions of research methods, a detailed description of main steps of a method may be used (Lebrun, 2011).

```
Count numbers from the abstract into NAC.
IF
  NAC == 0
THEN
  ABSTRACT_PRECISION_LOW = TRUE;
```

Listing 12: Precision in abstract

3.3 Introduction metrics

In sequential order, an introduction is, many times, the third section of a scientific paper. However, content-wise, introduction is the one to start the paper. Also, when it comes to allowed length, introduction is the first section that allows more verbose writing; Lebrun (2011) recommends at least 15 % of the whole paper length; Körner (2008) would keep it shorter than two-thirds of the length of Results section. As such, the purpose of an introduction is to *introduce* the reader to the topic and research, the paper describes (Lebrun, 2011; Alley, 1996; Körner, 2008). Generally it is recommended that an introduction would consist of these sections (Rosenfeldt et al., 2000; Kurmis, 2003; Körner, 2008; Singer and Hollander, 2009; Moreira et al., 2011; Alley, 1996; Lebrun, 2011): background to the research, importance of the research (justification), methodology used in research, and section describing the hypothesis and aims of research.

Lebrun (2011) lists the following attributes that qualify introduction: *mindful* (provides sufficient context to familiarize the reader and thus reduce the knowledge gap, and uses appropriate expressions especially when describing the work of others), *story-like* (answers questions raised by the title and abstract and uses active, personal voice), *authoritative* (accurate expressions and factual comparisons), *complete* (sections describing issues from the list of previous paragraph) and *concise* (fast, strong start and no excessive details). The following sections address these qualities and provide fur-

ther explanation and metrics for measurements.

Conciseness and Completeness

For many scientists, writing the introduction is considered “a necessary evil”, a task that has to be completed because introduction is one of the expected and required standard parts of a scientific paper (Lebrun, 2011). Thus, many keep the introduction brief. Conciseness, in many cases (such as presented in the previous sections and the ones following in this section), is justified and may ease the reader’s cognitive burden (Daneman and Carpenter, 1980; Daneman and Merikle, 1996). The downside for conciseness is that it may cause lack of detail and missing important elements, therefore causing incompleteness. In case of introduction, incompleteness may prevent non-expert readers to get enough introductive background information to fill their knowledge gaps. The larger the knowledge gap, the more difficulties the reader has in understanding the topic of paper and the less motivated he or she is in reading further. Also, according to Lebrun (2011) and Eisenhart (2002), not all reviewers are experts of the field or topic they review papers for, meaning that not only can incompleteness in an introduction cause a reader stop reading further, but also a reviewer reject the paper and prevent paper from being published. Despite this, there are places in introduction where concise writing style may benefit: at the beginning of an introduction and when describing details.

The beginning of an introduction should get directly to the point Lebrun (2011). After reading the title and abstract, the two filtering mechanics, the reader has decided to read further. He or she is interested and expects to find details expanding title and abstract, and background information to frame the context. A metric by Lebrun (2011) in Listing 13 addresses this issue by detecting possible “false starts”, in which the writer has decided to warm up the topic by providing unnecessarily general background information. The false starts reduce conciseness and delay the reader from advancing to more prominent matters, such as the direct context or impact of contribution. The metric asks the writer to identify the first sentence in introduction that contains uncommon knowledge for the non-expert reader. Sentences containing common knowledge even for the non-experts are considered to be potentially unnecessary.

```
MANUAL: User defines the first sentence in the introduction
    having uncommon knowledge for the non-expert reader of the
```

```
    paper.

IF
  the first sentence in the introduction does NOT have uncommon
     knowledge
THEN
  STRONG_START = TRUE;
ELSE
  STRONG_START = FALSE;
```

Listing 13: Beginning the introduction

The ending of an introduction, likewise, should be carefully considered, and contain description of the expected impact of contribution (Lebrun, 2011). Lebrun (2011) identifies six different possible endings for an introduction:

1. Ending describing the table of contents and upcoming sections
2. Ending describing impact of paper and contribution
3. Ending describing goal of paper and research
4. Ending describing methodology of work
5. Ending describing main results or anticipated results
6. Ending not describing any of the previous alternatives

The metric in Listing 14 lists outcomes of having each of the previous alternatives as an ending. Lebrun (2011) explains that the most ideal ending would be to have description of *expected impact of paper and contribution* at the end as it provides reader justification of the benefits and therefore motivates reader continue reading. Having a *table of contents* type of ending, even though generally recommended (Alley, 1996; Rosenfeldt et al., 2000), may not be necessary as a scientific paper is relatively short and structure can be seen fairly easily. Same applies for having *methodology description* as the ending: in a conventional scientific paper introduction is followed by the methodology section, thus causing unnecessary repetition. The mention of the *paper and research goals* should be placed to the beginning of an introduction instead of the end (Lebrun, 2011). Introducing the *main results of research* at the end does not necessarily motivate as much as stating the impact directly, because in order to realize the benefits, readers has to understand how to interpret the results. For non-experts this may be prove difficult. The last alternative, having something else as an ending, is un-

expected. Usually this should be avoided since readers expect to see certain elements in certain places.

```
MANUAL: User defines the purpose for the last paragraph in
    introduction: TC="table of contents for rest of paper
    covering upcoming headings", IP="impact of paper", GL="goal
    of paper", MT="methodology of work", RL="main result or
    anticipated result of research" or OT="other".

SWITCH purpose:
  CASE TC: CONVENTIONAL_ENDING = TRUE;
  CASE IP: EXPECTED_ENDING = TRUE;
  CASE GL: GOAL_SECTION_MISPLACED = TRUE;
  CASE MT: UNNECESSARY_INFORMATION_AT_END = TRUE;
  CASE RL: EXPECTED_ENDING = FALSE;
  CASE OT: CONVENTIONAL_ENDING = FALSE;
```

Listing 14: Ending the introduction

Having addressed the beginning and ending of an introduction, it is time to address the other issue mentioned at the first paragraph of this section: introduction *completeness*. To ensure all relevant information is included in the introduction, Lebrun (2011) advices to determine first the main question that is answered with stating the contribution, and then asking and answering the following four questions: "why the research is performed *now*?", "*why this topic* was chosen?", "why it was performed *this way*?" and "why should the *reader care*?". These are the questions a reader has in mind after reading the title and abstract. In the current (2012) implementation of SWAN, the completeness is measured by the following metric listed in Listing 15 (Lebrun, 2011). The metric compares and calculates percentage of introduction word count versus the word count of the full paper, excluding certain sections such as title, abstract and references, etc. If the size of introduction is below 5 % of the whole paper, the introduction would be better suitable for a letter than full scientific paper. An introduction below 15 % may indicate that not all expected sections were included into introduction, thus making the introduction feel incomplete.

```
Get the full text (exclude title, abstract, references, figures
    and their associated captions) and calculate words FWC.
    Calculate also the words in introduction IWC. Calculate
    percentage P: IWC / FWC * 100.
```

```
IF
  P < N%
THEN
  INTRODUCTION_SHORT = TRUE;
ELSEIF
  P < M%
THEN
  INTRODUCTION_SHORT_NOT_COMPLETE = TRUE;
ELSE
  INTRODUCTION_LENGTH_GOOD = TRUE;
WHERE (in the current implementation): N=5, M=15;
```

Listing 15: Introduction length

Appropriateness and Accuracy

The background and related work sections of an introduction handle the work of other researchers. A common trap, according to Lebrun (2011), is to use expressions that understate the work of others and on the other hand overstate one's own work when making comparisons between the past work and the contribution of one's paper. Understating other's work may also be unintentional, and happen with an imprudent choice of words. Particular caution should be used when selecting adjectives, verbs and adverbs to descriptions; some expressions are judgmental that make claims without providing evidence to support the claim (Lebrun, 2011). Some examples of such expressions are the adjectives "slow", "not reliable", "naive" and the verbs "fail", "not able to". A more complete list of judgmental expressions is seen in Appendix 1. A metric for searching such expressions is described in Listing 16. The metric processes through words in introduction and compares them with a list of aforementioned judgmental expressions. Lebrun (2011) recommends that such found expressions to be considered and rewritten to avoid unjustified judgments: instead of judging one might be better off with e.g. stating agreement or disagreement between the results, using facts and numbers that have objective nature and quoting papers that support one's own results.

```
Process through words in introduction. Search for words in list
   JGMWL indicating judgmental expressions. Search ignores word
   cases.

IF
```

```
  judgmental expressions in introduction:
THEN
  INTRODUCTION_HAS_JUDGMENTAL_EXPRESSIONS = TRUE;
WHERE (in the current implementation): JGMWL={"fail", "fails",
   "failed", "suffer", "slow", "limited", "unreliable", ... (see
   Appendix 1 for the full list)};
```

Listing 16: Judgmental words in introduction

Another perspective to the same issue is to overstate one's own contribution. These overstating expressions, like judgmental expressions, make claims without justification. These are the words such as "absolutely", "acute", "certainly" and "definite" (Lebrun, 2011). The exaggeration may cast disbelief into the minds of readers and reviewers – even to the point that also the well justified facts presented later in the paper are questioned (Lebrun, 2011). The probability of the paper to be published decreases as the reviewers' doubts increase and even if the work is published, the readers may decide not to trust the findings in the paper. Likewise with judgmental expressions, a metric by Lebrun (2011) exists (Listing 17) that processes and matches overstatements from the introduction. Found overstatements are recommended to be replaced with other expressions. A more complete list of expressions that overstate can be seen in Appendix 1.

```
Process through words in introduction. Search for expressions in
   list OVRWL indicating overstatements. Search ignores word
   cases.

IF
  overstatemental expressions in introduction:
THEN
  INTRODUCTION_HAS_OVERSTATEMENTS = TRUE;
WHERE (in the current implementation): OVRWL={"absolutely",
   "absolute", "abundantly", "acute", "acutely", ... (see
   Appendix 1 for the full list)};
```

Listing 17: Overstatements in introduction

Accuracy of expressions is also one factor impacting the credibility. Imprecise expressions may suggest or cast doubts that the writer possesses only superficial domain knowledge, therefore putting the value of contribution into doubt as well (Lebrun, 2011). The lack of detail also lowers the attractiveness of an introduction (Lebrun,

2011). The metric by Lebrun (2011) searches such expressions from the text: it uses a list of imprecise words and expressions to match words from an introduction (Listing 18). The list contains expressions such as “typically”, “overall” and “commonly”. Appendix 1 lists more such expressions that are used in the current implementation of SWAN. In addition to imprecise expressions, also imprecise references lower the accuracy of text. The imprecision may come in a form of insufficient amount of familiarization of source materials, careless reference placement, and/or grouped references (Lebrun, 2011). The inaccurate reference placement refers to situations when the reader can not be sure of which reference is used to support which claim. The reference should come immediately after the mention to avoid these problems (Council of Science Editors, 2006; Lebrun, 2011). The grouped references (e.g. [1,2,3]) decrease the accuracy of referencing as a claim can not be unambiguously traced to a specific reference; it also may suggest the writer has conducted hasty research (Lebrun, 2011).

```
searchImpreciseExpressions(introduction):
  Process through words in introduction. Search for words in
     list IMPWL indicating imprecise expressions. Search ignores
     word cases.
  IF
    imprecise expressions in introduction:
  THEN
    INTRODUCTION_PRECISION_LOW = TRUE;
  WHERE (in the current implementation): IMPWL={"typically",
     "generally", "overall", "commonly", "can", "may", ... (see
     Appendix 1 for the full list)};

searchImpreciseReferences(introduction):
  Search references from introduction and count those that are
     grouped together (i.e. [1,2,3]).
  IF
    count of grouped references in introduction > N
  THEN
    IMPRECISE_REFERENCES = TRUE;
  WHERE (in the current implementation): N=1;
```

Listing 18: Precision in introduction

Attractiveness

Attractiveness of a writing impacts on how motivated the reader remains as he or she reads further on. Since the title and abstract have already revealed the essence of the paper, the reader must be kept motivated by attractive writing. In introduction, attractiveness is affected by the following: usage of visuals and questions, sentence voices and pronouns, and sentence and phrase lengths and length variations. Also transitions between sentences (sentence progression fluidity) affect how easy it is to follow the text and thus how attractive the text will be seemed.

Visuals and questions are a good way to variate the writing and focus reader attention (Lebrun, 2011; Alley, 1996). A visual attracts attention and provides a wordless way to inform the reader; they provide support for the writing. Having visuals in the in introduction increases the attractiveness and motivates the reader. The Listing 19 includes metric for searching hints of visual usage from a text (Lebrun, 2011). Having no visuals makes the text unappealing. The metric does not, however, take a stance on the quality of the visual; it merely looks for references for visuals.

```
searchVisualsUsage(introduction):
  Search introduction for clues of visuals usage. Use list VL of
     words indicating visuals usage to be used in search. Search
     ignores word cases.
  IF
    occurrences of words in VL in introduction
  THEN
    INTRODUCTION_APPEALING = FALSE;
  WHERE (in the current implementation): VL={"fig", "figure",
     "table"};
```

Listing 19: Usage of visuals in introduction

According to Lebrun (2011), questions create suspension and get reader's attention. When a question is asked in a text, it refocuses reader's mind and makes him or her want to know the answer to that question. A question also sets the topic of the paragraph and gives direction to ideas. The question does not need be direct: also implicit questions can accomplish the same effect. The following presents different kind of statements that, according to Lebrun (2011), raise questions:

Direct question "What would be, given these requirements, the best way to achieve

the aim?"

Indirect question Questions asked by the writer: "Given these requirements, we wondered the best way to achieve the aim."

Announcing unexpected findings Expressions that announce findings that raise questions: "Surprisingly, our data showed an increase of 15 % in ..."

Not-yet-justified adjectival claim A claim that raises question before providing answer: "The results of our study were more complete than ..."

Negative statement Statement contrasting non-working or unimportant issues with what are: "The amount of objects in the stack list is unimportant, whereas the size of object is what matters."

Announcement of change "The progression of the development of technology had been slow, but this was about to change."

Provocative statement Bold statements and claims for what readers' yearn justification: "World Wide Web will die in 5 years."

Values in visuals Values that vary from the standard and thus attract attention and explanation.

Antagonistic claims Statements that make comparisons and claims using words such as "whereas", "contradiction".

Roadblock Stating the inability to compare findings due to different methods or results.

The current implementation of SWAN, however, does not identify the implicit questions. Instead it relies on identifying only the direct questions. The metric by Lebrun (2011) is described in Listing 20 and simply iterates through the text in introduction and tries to find question marks. To increase attractiveness, the introduction should include questions.

```
searchQuestionUsage(introduction):
  Count questions (question marks) from introduction into IQC.
  IF
    IQC == 0
  THEN
    INTRODUCTION_ATTRACTIVE = FALSE;
```

Listing 20: Usage questions in introduction

The choice of pronoun usage and sentence voices change the writing style. Many scientists avoid personal pronouns and active sentence voices and instead prefer passive

and impersonal writing (Alley, 1996; Lebrun, 2011). According to Alley (1996) this preference originates from scientists' misconceptions of how scientific writing should be. Some also argue that passive voice increases paper's authoritativeness (Lebrun, 2011). However, passive and impersonal voice both slow writing and reading, and may lead to unnatural wording (Alley, 1996). Using active voice and personal pronouns instead may increase clarity and attractiveness as it 1) makes identifying authors contribution from the others easier, 2) reinforces reader motivation with a welcoming voice that active and personal writing establishes and 3) reduces ambiguity (Lebrun, 2011). Alley (1996) also states that the nouns and verbs used in active voice are strong and provide both anchors and momentum that accomplish fluid writing. According to Lebrun (2011), active voice is recommended for introduction; Alley (1996) states that active voice suits everywhere as long as the emphasis is on the study and not on the author. A style manual for NASA Langley Research Center (McCaskill, 1998) also recommends using active voice as it increases text conciseness.

Current implementation of SWAN includes metrics by Lebrun (2011) to examine the usage of personal pronouns and sentence voices in a text. The first of these metrics, used to find personal pronouns, is described in Listing 21. It iterates through the words in introduction and searches words that indicate usage of personal pronouns. Having under two such occurrences may indicate impersonal writing style.

```
Search introduction for of personal pronoun usage. Use list PNL
    of words indicating personal pronoun usage to be used in
    search. Search ignores word cases.

IF
  occurrences of words in PNL in introduction < N
THEN
  INTRODUCTION_IMPERSONAL = TRUE;
WHERE (in the current implementation): PNL={"we", "our"}, N=2;
```

Listing 21: Usage of personal pronouns in introduction

Listing 22 describes the other of the aforementioned metrics; the one used to determine the sentence voices. The metric relies upon natural language processing (NLP) tools, which in the current implementation are provided by the Stanford Parser library (Klein and Manning, 2003). The library is able to determine grammatical dependencies and relations (De Marneffe and Manning, 2008a) and thus separate passive expressions

from active ones. Section 4.3 addresses the topic in more detail. The metric counts the passive sentences detected by Stanford Parser and compares it with the active sentence count. If there are more passive sentences than 50 % (in the current implementation) of the count of active sentences, the introduction is judged to be in passive voice.

```
Count IS, total amount of sentences in introduction.
Count NP, introduction sentences written with passive voice.
Count NA, introduction sentences written with active voice: NA =
    IS - NP

isSentencePassive(sentence):
  dependencies = StanfordNLP.getTypedDependencies(sentence)
  FOR dependency in dependencies:
    relation = dependency.relation
    IF relation in DL
      SENTENCE_PASSIVE = TRUE;
      BREAK;
    ELSE
      SENTENCE_PASSIVE = FALSE;
      BREAK;

IF
  NP > a * NA
THEN
  INTRODUCTION_IN_PASSIVE = TRUE;
WHERE (in the current implementation): DL={"auxpass",
    "csubjpass", "nsubjpass"}, a=0.50;
```

Listing 22: Sentence voices in introduction

Reading long and complex phrases and sentences consume much energy from the reader (Lebrun, 2011); they burden one's cognitive abilities (Gopen and Swan, 1990; Daneman and Carpenter, 1980; Daneman and Merikle, 1996). The longer the sentences are, the more likely it is that they are full of redundancies and writing zeroes making sentences complex; such "fat writing" slows down writing as well (Alley, 1996). By increasing conciseness, phrases and sentences become easier to understand and faster to read. Both Lebrun (2011) and Alley (1996) recommend keeping the average sentence length equal or below 20 words. Lebrun (2011) also goes further and makes recommendations of phrase lengths inside sentences (average length equal or below 8 words). However, according to Gopen (2004) the length by itself does not necessarily

make a sentence hard to read, but rather hard to write so that it remains readable. The metric in Listing 23 addresses both sentence and phrase length recommendations.

```
calculateSentenceLength(introduction):
  Iterate through sentences in introduction. Count sentence
     lengths in words. Calculate average AVG_S of sentence
     lengths.
  IF
    AVG_S > N_S
  THEN
    INTRODUCTION_LONG_SENTENCES = TRUE;
  WHERE (in the current implementation): N_S=20;

calculatePhraseLength(introduction):
  Iterate through phrases in introduction. Count phrase lengths
     in words. Calculate average AVG_P of phrase lengths.

  Phrase segment is sentence or a part of sentence starting and
     ending with a character in P.
  IF
    AVG_P > N_P
  THEN
    INTRODUCTION_LONG_PHRASES = TRUE;
  WHERE (in the current implementation): P={'.', ',', ';', '!',
     '?', ':'}, N_P=8;
```

Listing 23: Introduction sentence and phrase length

The length is not the only factor that makes writing dull in the eyes of the reader: also phrase and sentence *variations* play a role in this (Alley, 1996; Lebrun, 2011). Small variation between phrase and sentence word counts or repeatance of similar sentence patterns results into boring writing and decreases both fluidity and attractiveness. Alley (1996) mentions several ways to begin sentences that may be used to vary sentence patterns (e.g. subject-verb, verb phrase and infinitive phrase patterns). As a guideline, sentence lengths and patterns should be varied every two or three sentences (Alley, 1996; Lebrun, 2011). Gopen (2004), however, does not recommend doing so without thinking the function of the sentence: instead, varying sentence structures should happen in direct relation to the function they perform as a unit of discourse. The metric in Listing 24 by Lebrun (2011) examines the input text by calculating average and standard deviation from the sentence/phrase word counts. For sentences, the standard

deviation is recommended to be over 25 % of the average sentence length plus 3 words. For phrases, the standard deviation should be over 4 words.

```
calculateSentenceVariation(introduction):
  Iterate through sentences in introduction. Count sentence
     lengths in words. Calculate average sentence word length
     AVG and standard deviation STDEV_S.
  IF
    STDEV_S < AVG / N_S
  THEN
    INTRODUCTION_SENTENCES_ATTRACTIVE = FALSE;
  IF
    STDEV_S > AVG / N_S + M
  THEN
    INTRODUCTION_SENTENCES_ATTRACTIVE = TRUE;
  WHERE (in the current implementation): N_S=4, M=3;

calculatePhraseVariation(introduction):
  Iterate through phrases in introduction. Count phrase lengths
     in words. Calculate standard deviation STDEV_P of phrase
     lengths.

  Phrase segment is sentence or a part of sentence starting and
     ending with a character in P.
  IF
    STDEV_P < N_P
  THEN
    INTRODUCTION_PHRASES_ATTRACTIVE = FALSE;
  WHERE (in the current implementation): P={'.', ',', ';', '!',
     '?', ':'}, N_P=4;
```

Listing 24: Introduction sentence and phrase length variation

Transition words are expressions (e.g. “on the other hand”, “moreover”) used to move from sentence to sentence or paragraph to paragraph (Lebrun, 2011). These words link topic from sentence A to sentence B. However, the connection they establish is often artificial and decrease writing’s fluidity (Lebrun, 2011). Thus, in most cases, the transition words should be replaced with phrases expressing implicit progression (e.g. sequential step) (Lebrun, 2011). The metric by Lebrun (2011) in Listing 25 processes through the words in introduction and searches for transition words such as “on the other hand”, “also”. Appendix 1 contains the complete list of transitional expressions.

In some cases, however, use of transition words is justified: such is the case when connecting two independent sentences not sharing a common topic (Lebrun, 2011).

```
Process through words in Introduction. Search for expressions in
    list TRWL used in transitions. Search ignores word cases.
IF
  transition expressions in Introduction:
THEN
  INTRODUCTION_HAS_TRANSITIONAL_EXPRESSIONS = TRUE;
WHERE (in the current implementation): TRWL={"On the other
    hand,", "And,", "Also,",  "Moreover,", ... (see Appendix 1
    for the full list)};
```

Listing 25: Transition words in introduction

3.4 Conclusions metrics

Conclusions section is one of the last ones in a scientific paper – as such, it has the role of *concluding* what has been brought up earlier in the paper (Lebrun, 2011; Alley, 1996). Conclusions section may not necessarily be a distinct section as the title, abstract or introduction: depending on journal, conclusions may not have a distinct heading or section but they are presented in part of discussion section (Lebrun, 2011; Ortinau, 2011; Körner, 2008; Katz, 2009). Regardless, a paper needs some form of conclusion (Montgomery, 2003; Lebrun, 2011). According to Lebrun (2011), conclusion statements should be written with assurance and in positive voice for not unmotivating the readers that has decided the research is worth of their time (Lebrun, 2011). Montgomery (2003) suggests conclusions section to be a return to the research topic introduced at the beginning of paper with an additional statement of what new has been added (refocusement of contribution). Also, according to Montgomery (2003), conclusions section should be written with the most detailed information at the beginning and the broadest statements at the end (as a mirror to the introduction).

For a reader, the conclusions section brings a contrast between the *pre-contribution* (background of research and research field) from introduction and *post-contribution* (research results, limitations and future work) mentioned in conclusions (Lebrun, 2011). For the writer, conclusions is a possibility to polish contribution and underline its importance for the reader, as well as announcing and proposing future research.

Lebrun (2011) recommends that a conclusion should have the following qualities: be *positively charged* for motivating the reader, be *predictable* (contain nothing that has not been mentioned or hinted previously and contain all expected sections), be *concise* (bring closure and mention future work in concise manner), and be *coherent* with title, discussion and introduction. These qualities, with metrics to assess them, are addressed in more detail in the following sections.

Complete conclusions

Conclusions section brings closure to the issues brought up in introduction, discussion and other parts of scientific paper (Lebrun, 2011; Alley, 1996; Montgomery, 2003). It should repeat the key points of the paper, but nothing that has not been brought up in other sections: whereas in abstract everything is new to the reader, in conclusions nothing is (Lebrun, 2011; Alley, 1996). However, conclusions should not be a mere compilation of sentences from other sections (Lebrun, 2011); instead, conclusions should contain implications of what has been presented in the paper (Lebrun, 2011; Kurmis, 2003). These implications should contain mention of impact of research and its results, scope and limitations in when research hypothesis works or does not work, and potential future work (Lebrun, 2011; Ortinau, 2011).

An indicator for the completeness of a conclusions section is its length (Alley, 1996; Lebrun, 2011). Whereas length by itself does not necessarily equate with completeness, it is, however, a fairly good pointer: the more words has been used in a section, the better probability there is that the section contains more information; vice versa, the fewer words there is, the less probable it is that all *necessary* information fits to the section. What would, then, be a good length for a conclusions section? According to Alley (1996) it depends on the type of paper: in a short paper even one sentence may suffice; in typical scientific papers, the conclusions should be at least as long as the abstract. Lebrun (2011) also agrees with this. Also, a metric by Lebrun (2011) described in Listing 26 uses the same recommended length to determine whether a conclusions has sufficient length. The metric counts words from the paper's abstract and conclusions sections and compares the counts. Conclusions sections under the length of an abstract have the risk of not being developed enough to contain all necessary information and therefore not having a satisfactory conclusion for the reader (Lebrun, 2011).

```
Count words from conclusions into CWC. Count words from abstract
```

```
    into AWC.
IF
  CWC / AWC * 100 < N%
THEN
  CONCLUSION_SHORT = TRUE;
ELSE
  CONCLUSION_SHORT = FALSE;
WHERE (in the current implementation): N=100;
```

Listing 26: Conclusions length

Besides, and related to, the length of conclusions section, completeness or lack of it can be determined by making assessments of what information conclusions section contains. As mentioned earlier, conclusions should contain mention of 1) impact and results of a research, 2) scope and limitations in which research hypothesis works, and 3) potential future work. Of these three, 1) and 2) are used to bring closure to the current research; 3) on the other hand, can be used to offer a glimpse of what could come next (Lebrun, 2011; Alley, 1996). A conclusions that contains these three, therefore has a better probability of satisfying the needs of a reader. A metric that would assess whether these three parts are in a given conclusions section, could therefore determine if conclusions section is complete. However, the current implementation of SWAN and its metrics, does not (at least at the moment) consider all three parts, but instead focuses on the part 3), the part containing mention of future work.

The future work section can contain guidelines, directions and plans for the next stage of research, and give a signal to the readers that they should stay tuned for the coming (Lebrun, 2011). It can also be used to address some limitations in the current hypothesis to convince readers that the limitations are not lasting and will be corrected in the future (Lebrun, 2011). Listing 27 contains a metric by Lebrun (2011) for assessing whether future work is mentioned in conclusions. The metric uses a list of expressions indicating future work (e.g. “future”, “intention”) and compares each word in conclusions with the list. Lack of occurrences indicate that future work section is missing and thus may make conclusions feel incomplete and leave readers unsatisfied.

```
Count number of future work expressions (FWE) in conclusions.
    The comparison between conclusions section words and words in
    FWE is case insensitive.
IF
  future work expressions NOT found from conclusions section:
```

```
THEN
  FUTURE_WORK_MISSING = TRUE;
WHERE (in the current implementation): FWE={"future", "intend",
   "intention", "plan", "limit", "will", "further", "expect",
   "anticipate", "project to"};
```

Listing 27: Future work section

Positive and attractive conclusions

Although the role for attracting and motivating readers to read further is usually given to introduction section (see Section 3.3), also conclusions section should be attractive and motivating for the sake of non-linear nature of scientific reading. Non-linear reading means that readers may start reading the paper from the abstract, decide to skip introduction section and jump directly into conclusions section; therefore the conclusions section is responsible for attracting and motivating reader to read further (Lebrun, 2011). Also, for the same reason, an abstract and conclusions should not be too similar: repeating the same or similar sentences in abstract and conclusions or having otherwise too similar sentence patterns, causes conclusions feel a mere repeat of abstract and unmotivate readers (Lebrun, 2011).

One good way to differentiate abstract and conclusions is, according to Lebrun (2011), to use distinct verb tenses: *dynamic present verb tense* in an abstract (see Section 3.2) and *past tense* to signify the end in a conclusions. One exception for the past tense recommendation are the unquestionable facts in text: they should be presented in present tense because present tense reinforces contribution when used with facts (Lebrun, 2011). Day (1998) also recommends using present tense when referring to established facts (previously published information), and using past tense when describing results of the current research.

A metric by Lebrun (2011) described in Listing 28 addresses the verb tense issue. The metric iterates through the sentences in conclusions and makes counts for number of sentences and different verb tenses. A lack of verbs in present tense may indicate unconvincing presentation for achievements and facts, because other tenses have been used. Otherwise conclusions section should be written in past or present perfect tenses: according to Lebrun (2011), the amount of these tenses should be at least 70 % of the amount of sentences in conclusions section; having less than this may make conclu-

sions and abstract feel too similar as both consists of the same verb tenses.

```
Iterate through sentences in Conclusions. Count NCS, the number
    of sentences, CVPR; the number of verbs in present tense;
    CVP, the number of verbs in past tense; and CVPP, the number
    of verbs in present perfect tense.

IF
  CVPR == 0
THEN
  ACHIEVEMENTS_PRESENTED_CONVINCINGLY = FALSE;
IF
  CVP + CVPP < NCS * N
THEN
  CONCLUSION_NOT_IN_PAST_TENSE = TRUE;
WHERE (in the current implementation): N=0.7;
```

Listing 28: Tenses in conclusions

3.5 Structure metrics

Scientific paper's structure, consisting of upper and lower level headings and their respective body text sections, represents research divided into logical parts (Alley, 1996). A common structure follows *IMRaD* organization, which consists of Introduction, Methods, Results and Discussion sections (Alley, 1996; Day, 1998). In addition to this, a paper has a main title and may have an abstract and a separate conclusions section, depending on scientific field and journal (Lebrun, 2011); some journals guide to integrate conclusions into Discussion section (see Section 3.4). Also, a paper may include supplementary sections such as "acknowledgments", "references" and appendices (Kurmis, 2003). Together these sections form the main level structure with standard headings; the middle and lower level sections and their headings, on the other hand, vary from paper to paper, because they contain the unique contributive information of the paper (Lebrun, 2011).

A structure should help reader to navigate inside the paper and focus on the sections he or she is most interested in (Lebrun, 2011; Alley, 1996). A reader should also get a clear picture of the contents of paper after examining the structure (Lebrun, 2011). For a writer structure can be used to emphasize the contribution of a paper by repeating

keywords from the title and abstract in the structure headings (Lebrun, 2011). Because a reader typically remembers only 10 % to 20 % of what they have read, repeating the most important issues helps them in memorizing, and also emphasize what is important (Alley, 1996). The structure also helps dividing paper into informative and logical sections (Lebrun, 2011; Alley, 1996); Alley (1996) mentions different strategies to be used in organizing information (some words about this later).

A structure acts as a skeleton to a paper by supporting its parts (Lebrun, 2011). Lebrun (2011) has four principles that a good structure follows:

1. Contribution guides the shape of structure
2. Sections containing contribution are grouped
3. Main title is connected to structure, and vice versa
4. Structure is logical and tells a clear story

Principles 1 to 3 are directly connected to the contribution of the paper: structure accommodates to the contribution by having it shaped so that contribution is emphasized. Principle 4 also concerns contribution, but focuses more on how logical and clear the structure should be. These principles guide structure towards qualities of a good structure (Lebrun, 2011): a structure should be *informative*, *tied* to title and abstract, *logical*, *consistent*, *clear*, and *concise*. The following sections target the mentioned principles and qualities more closely and provide metrics for assessment.

Contribution shaped structure

Contribution shapes structure in many ways: it both determines the outline (number of heading levels, and headings in each level), and how much information should be included in sections. In scientific papers, the amount of detail usually increases every heading level (Davis et al., 2013). Thus the most detailed information is usually found from the lowest structure levels. Because the contribution should contain the unique information and be most detailed, it should also be found from the lowest level (Alley, 1996; Lebrun, 2011). The high detail level usually comes in hand with text length: the longer the text, the more detailed information it potentially contains (Alley, 1996). The current version of SWAN contains two related metrics by Lebrun (2011) that we now describe.

The first metric in Listing 29 examines paper structure for sections reflecting contribution (determined by user). Contributive section not in the lowest structure level may indicate that contribution is not detailed enough, and a secondary section such as section describing background to the research has too great detail level (Lebrun, 2011). Figure 2 illustrates this by providing two example structures, from which the first (a) does not contain contribution in its lowest level; example (b) on the other hand, does.

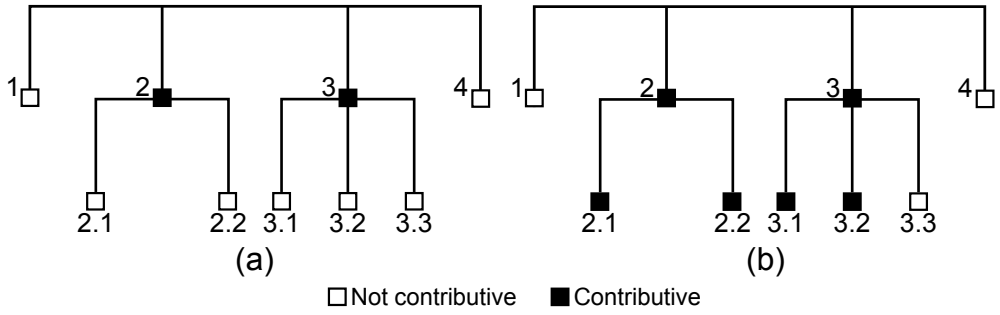

Figure 2: Contribution should be found from the deepest structure level. In the example structure (a) structure has deeper-level sections (e.g. 2.1 and 2.2) than the ones reflecting contribution (2 and 3). In example (b) contribution is found in e.g. Section 2.1, which is one of sections in the deepest level of structure. The example structure (b) is thus preferred. (Lebrun, 2011)

```
MANUAL: The paper structure is determined and sections
    reflecting core of contribution are marked.

IF
  deepest level section does NOT reflect core of contribution
THEN
  CONTRIBUTIVE_SECTION_HAS_ENOUGH_DETAIL = FALSE;
```

Listing 29: Core of contribution depth

The second metric, in Listing 30, examines detail level of sections. It uses section word count as an indicator for information detailness. The metric compares word counts of sections reflecting core of contribution to other sections and to the whole paper. Contributive sections should consist of 50 % to 75 % words of the whole paper. Percentages under 50 % may indicate that the paper consists of larger background than required. On the other hand, if contributive sections take over 75 % of the whole paper, there may not be enough background information, and thus reader may suffer from too large a knowledge gap to get full benefits from the contribution. A single contributive section should consist up to 30 % words of the whole paper and should be reorganized to smaller sections if necessary. According to Alley (1996), there are no absolute

values for section lengths and that they depend on research and audience. However, he nevertheless recommends dividing sections exceeding ten paragraphs into multiple sections to allow the reader a pause. A single section without contribution should be longer than 5 % of the whole paper to avoid having too many sections in the paper (Lebrun, 2011). Since headings create pauses in reading, having too many headings also interrupt reader's thoughts and tire them (Alley, 1996).

```
MANUAL: paper structure is determined and sections reflecting
    core of contribution are marked.

Calculate word counts for sections and subsections. Count total
    words TOTAL. Calculate word counts for sections reflecting
    core of contribution. Count total words in contributive
    sections CONTRIBUTIVE_TOTAL. Count words in largest
    contributive section LARGEST_CONTRIBUTIVE. Count words in
    smallest non-standard section SMALLEST_SECTION.

IF
  CONTRIBUTIVE_TOTAL / TOTAL * 100 < CONTRIBUTIVE_MIN_P%
THEN
  CONTRIBUTIVE_SECTIONS_TOO_SMALL = TRUE;
ELSEIF
  CONTRIBUTIVE_TOTAL / TOTAL * 100 < CONTRIBUTIVE_MAX_P%
THEN
  CONTRIBUTIVE_SECTIONS_OF_GOOD_LENGTH = TRUE;
ELSEIF
  CONTRIBUTIVE_TOTAL / TOTAL * 100 > MAX_P%
THEN
  CONTRIBUTIVE_SECTIONS_TOO_LARGE = TRUE;

IF
  LARGEST_CONTRIBUTIVE / TOTAL * 100 > SECTION_MAX_P%
THEN
  TOO_LARGE_SECTION = TRUE;

IF
  SMALLEST_SECTION / TOTAL * 100 < SECTION_MIN_P%
THEN
  TOO_SMALL_SECTION = TRUE;

WHERE (in the current implementation): MIN_P=50, MAX_P=75,
```

```
SECTION_MAX_P=30, SECTION_MIN_P=5;
```

Listing 30: Word distribution

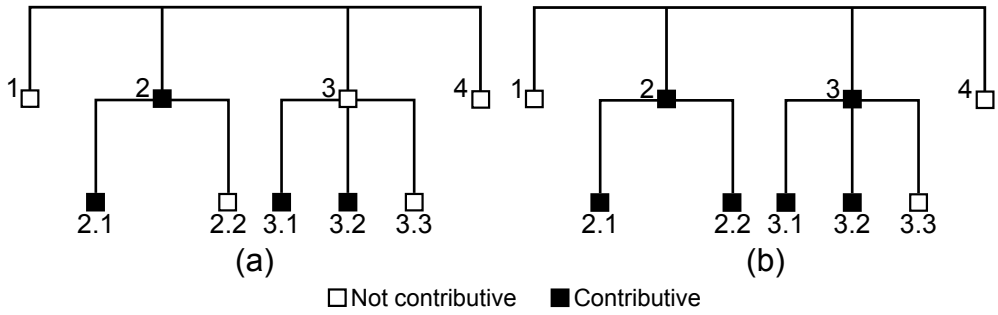

Figure 3: Sections reflecting core of contribution should be in one group. In example structure (a) there are two groups [2, 2.1] and [3.1, 3.2], causing scattered contribution. In (b) contribution is focused in one group [2, 2.1, 2.2, 3, 3.1, 3.2]. Thus, the (b) structure is better, than (a). (Lebrun, 2011)

The second principle by Lebrun (2011) states that sections reflecting contribution should be grouped. That is, paper should be written and organized so that once paper starts handling the contribution, it lasts uninterrupted until all of that, what will be told about contribution, is told. Contributive sections forming a single group is a indication of well identified contribution (Lebrun, 2011). On the other hand, having contribution scattered into multiple groups causes lack of focus, unity and conciseness. The paper may have multiple contributions, in which case it is recommended to write more than one paper (one paper for one contribution). It might also be that the author has not been able to identify contribution well enough (Lebrun, 2011). This issue is illustrated in Figure 3. The corresponding metric by Lebrun (2011) is described in Listing 31. The metric sorts structure sections sequentially, and forms groups of uninterrupted sections reflecting contribution. The determination of which sections reflect contribution is done by the user.

```
MANUAL: paper structure is determined and sections reflecting
   core of contribution are marked.

Traverse the paper structure in sequential order (1, 1.1, 1.2,
   2, 2.1, ... N.M) and form groups of sections reflecting
   contribution. If there is a section not reflecting
   contribution, close current group, and start new group at
   next section that reflects.

IF
```

```
  there are more than one group reflecting contribution
THEN
  CONTRIBUTION_SCATTERED = TRUE;
```

Listing 31: Contribution scatterance

The last principle connected to contribution relates to connection between paper's main title and structure. The title states contribution via contribution keywords (see Section 3.1). The structure, on the other hand, helps reader navigate the paper and identify contribution (Lebrun, 2011). This is accomplished with a tight connection between title and structure headings. A metric by Lebrun (2011) in Listing 32 describes this in more detail. In short, title and structure are tightly connected, when all title keywords reflecting contribution are found from structure headings, and there are no words in the headings of contributive structure sections that are not found from the title. The metric examines four possible cases:

1. Contributive title keywords are completely missing from headings
2. One or more, but not all, title keyword is missing from headings
3. Sections reflecting contribution have words in its heading not found from the title
4. Section reflecting contribution does not have any of the title keywords in its heading

In cases one, two and three all or some of the contributive title keywords are missing from structure headings or some words from headings are not present in the title. In these cases, there is a disconnection between title and structure. Either title or structure is imperfect (Lebrun, 2011). If, for example, structure is missing a title keyword, either the structure has a logical gap, or the title word is not important, and should be removed from the title. The author should consider which one, title or structure, reflects contribution better and make adjustments accordingly. In case four a contributive section and the main title are disconnected. The title may be imperfect and author should add words into it, or the contribution presented in the section is impossible to fit into title; in that case, the paper may have more than one contribution. A typical cause for disconnection, in all of these cases, is inconsistent use of words: either title or structure contains synonyms or acronyms, or has words in different detail level (Lebrun, 2011). Alley (1996) encourages creating headings with same principles as the paper's main title (see Section 3.1): with clarity and precision in mind.

```
MANUAL: The paper structure is determined and sections
    reflecting core of contribution are marked. Contributive
    title keywords are determined.

Compare contributive section headings with contributive title
    keywords. Comparisons are case insensitive and use stem form
    of words. Comparison ignores prepositions and articles, words
    in list IWL, punctuation and numbers.

IF
  no contributive title keywords found from contributive section
      headings
THEN
  CONTRIBUTIVE_TITLE_KEYWORDS_COMPLETELY_MISSING_FROM_HEADINGS =
      TRUE;
ELSEIF
  one or more contributive title keywords NOT found from
      contributive section headings
THEN
  CONTRIBUTIVE_TITLE_KEYWORDS_MISSING_FROM_HEADINGS = TRUE;

IF
  contributive section headings contain words not found from
      contributive title keywords
THEN
  CONTRIBUTIVE_HEADINGS_CONTAIN_WORDS_MISSING_FROM_TITLE = TRUE;

IF
  contributive section heading does not contain any contributive
      title keywords
THEN
  CONTRIBUTIVE_HEADING_MISSING_ALL_TITLE_KEYWORDS = TRUE;

WHERE (in the current implementation): IWL={"using", "based on"};
```

Listing 32: Structure words in title

Logical, informative and clear structure

A logical, clear and informative structure guides reader through the paper and allows him or her to concentrate on most interesting sections (Lebrun, 2011; Alley, 1996). The

structure is logical in a) how sections are divided, b) in which order the sections are, and c) how the section headings are titled (Lebrun, 2011; Alley, 1996). For the part c), Alley (1996) recommends using parallel section heading titles. This means that structure should not contain mixed verb phrase, noun phrase and full sentence headings, but consistently use only one type of phrases. For example, a structure consisting mainly of verb phrase headings such as "Formalizing quality metrics" should not be mixed with noun phrase headings such as "Implementation for Scientific Writing Assistant".

The a) and b) parts depend on what strategy is used to organize the paper (Alley, 1996). For example, for a paper that discusses time-line processes, a *chronological strategy* is suitable (Alley, 1996). In this strategy contents are divided into steps that follow chronological order. Other strategies are for example *spatial strategy* (structure follows physical shape of an object), *flow strategy* (structure follows flow of some variable through a system) and *comparison-contrast strategy* (structure consists of comparable issues). These strategies may help in constructing logical structure. Which strategy is the most appropriate, depends on topic and audience. However, regardless of the strategy, one way to test whether structure is logical, is to compare it with the abstract (Lebrun, 2011).

An abstract contains summary of the most prominent elements of the paper (see Section 3.2); in a sense it is also a summary of the paper's structure. Therefore for the structure to tell a logical and clear story, it should be connected to the abstract (Lebrun, 2011). In other words, keywords from abstract should be found from section headings and vice versa. The metric in Listing 33 is developed for this purpose: it compares words in abstract to words in section headings. The comparison is case insensitive, uses stemmed words, and excludes general words (such as "data" and "method"), auxiliary verbs (such as "could" and "shall"), prepositions, pronouns and numbers. Appendix 2 provides complete list of excluded general words and auxiliary verbs. If section heading contains a word not found in abstract, author should consider the following questions (Lebrun, 2011):

How significant the heading word is? If the word is significant enough to be used in structure, author should consider adding the word into abstract.

Does the disconnection occur because a synonym or an acronym is used? If the heading word is synonym or acronym to a word in abstract or vice versa, author should consider replacing it with the original word.

Is the heading word highly specialized? If the word is too specific to be used in an abstract and too specific for a non-expert reader to understand, author should consider replacing it with a more understandable word.

As the list above points, a missing abstract word does not necessary indicate problems in structure, but the author should also verify whether the abstract fulfills its purpose (see Section 3.2) and whether it is the structure that reflects contribution better (Lebrun, 2011).

```
MANUAL: The paper structure is determined.

Compare words in abstract to words in section headings.
    Comparison is case insensitive and uses stem form of words.
    Comparison ignores general words and auxiliary verbs GAVL,
    prepositions, pronouns and numbers.

IF
  section heading word not in abstract
THEN
  STRUCTURE_WORDS_MISSING_IN_ABSTRACT = TRUE;
WHERE (in the current implementation): GAVL={"data", "method",
    "could", "shall",... see Appendix 2 for the full list}
```

Listing 33: Structure words in abstract

Structure should be informative: only the standard headings (IMRaD and Conclusions) should contain non-informative words (Lebrun, 2011). Non-informative words are words such as “characterization”, “demonstration” and “simulation” that, by themselves, do not give specific hints of the contents. The metric in Listing 34 searches section headings for non-informative words. Headings containing *only* such words are usually disconnected from the other paper parts (Lebrun, 2011) and thus make structure illogical and uninformative. Using non-informative words in general also break the recommendations for concise writing (Alley, 1996; Lebrun, 2011).

```
MANUAL: The paper structure is determined.

Search section headings (all levels) for non-informative
    expressions NIEL. Comparison is case insensitive and
    considers also plural and gerund form of non-informative
    expressions. Search excludes standard sections such as
```

```
    introduction and conclusions; see Appendix 2 for the full
    list.

IF
  section heading contains expression from NIEL AND section
     heading contains nothing else
THEN
  STRUCTURE_HAS_NONINFORMATIVE_KEYWORDS = TRUE;
WHERE (in the current implementation): NIEL={"characterization",
    "demonstration", "simulation", ... see Appendix 2 for the
    full list}
```

Listing 34: Non-informative headings

A structure should also be clear to the reader and reveal its contents easily (Lebrun, 2011). The metric in Listing 35 searches structure headings (to the lowest heading level) for acronyms. An acronym in a heading without the original term is ambiguous (Lebrun, 2011; Barrass, 2002; Day, 1998). Firstly, acronyms are not unique: there can be multiple terms for the same acronym (Barrass, 2002). For example acronym "CGI" stands for both "computer-generated imagery" and "common gateway interface", which are both common terms in computer science. Secondly, terminology, and thus the acronym for the term, can also change altogether. When this happens, and the reader is only offered an acronym that references to the old term, it may leave readers knowing only the newer acronym, wondering (Day, 1998). Non-expert readers of the research field may also not be familiar with field specific acronyms. These issues may result in reduced clearness for the whole structure and make it hard for the reader to follow (Lebrun, 2011).

```
MANUAL: The paper structure is determined.

Search structure headings (all levels) for acronyms.

IF
  section heading contains an acronym
THEN
  STRUCTURE_HAS_ACRONYMS = TRUE;
```

Listing 35: Acronyms in structure headings and subheadings

3.6 Fluidity metrics

Fluidity, in general, refers to how easily a passage of text can be read. The many benefits of fluid text include faster reading, better reading comprehension, increased reader satisfaction, and better possibilities for having the text published (Lebrun, 2011; Gopen, 2004). Fluidity also decreases the amount of interpretations readers get from the text, thus increasing the possibility that readers will interpret text the way writer meant it to be interpreted. But, when is a passage of text fluid? The answer lies within those who interpret the text: the readers (Lebrun, 2011; Gopen, 2004).

Readers form interpretations while reading a text (Gopen, 2004). The interpretation is influenced by two factors (Gopen, 2004): 1) by reader's *background* (prior knowledge of the topic, culture, temperament, job, reading habits, etc.), and 2) by the *expectations* reader has regarding the text. Reader's background can be tried to be taken into account by avoiding culture-specific idioms and by providing sufficient background information about the topic (Lebrun, 2011). The expectations regarding text (which are also influenced by reader's background), can be taken into account when writer becomes conscious of what expectations readers actually have; Gopen (2004) has developed *Reading Expectation Approach* (REA) for this purpose. Reading Expectation Approach is based on the general expectations a modern English reader has regarding certain structural positions and what substance they expect to find from those positions. With this information, writers can place material at positions where readers expect to find it. The general expectations readers have, are:

1. The placement of words within a sentence
2. The progression and links between sentences
3. The placement of sentences within a paragraph
4. The progression and links between paragraphs

The following sections will discuss these expectations. The main focus will be on the sentences, as they are also on the main focus in the fluidity metrics SWAN contains; however, a short discussion of expectations towards paragraphs and progression between them is also provided. After this, fluidity metrics based on these expectations and developed by Lebrun (2011), are discussed. These metrics can be used to automatically assess fluidity between a given set of sentences.

Expectations towards sentences

We first consider the smaller of the mentioned units of text: the sentences. But before we discuss expectations in more detail, we will, as a preparation, take a step back, and first consider some grammatical issues regarding sentences. A typical English sentence structure consists of a subject, a verb and a complement, as can be seen from Table 1 (Gopen, 2004). The subject acts as an agent, the performer of an action (in active voice sentences; passive voice sentences does not necessarily have explicit agents). Action, in turn, is articulated by the verb. The complement is affected by the action and indicates the goal of the sentence.

Table 1: Default sentence structure in English (Gopen, 2004)

Structure	Subject	Verb	Complement	Fixed
Substance	Agent	Action	Goal	Movable

The structure is relatively fixed and therefore, the varying factor is the content of substance, i.e. the meaning (Gopen, 2004). This information leads us to first three reader expectations concerning placement of words within a sentence (Gopen, 2004):

- Readers usually expect the action of sentence to be articulated by verb, and verb to express action
- Readers expect every subject to be followed almost immediately by its verb
- Readers expect certain type of substance to come in certain structural positions (Table 1)

Thus, expressing action with a word belonging to some other word class than verb, violates reader expectations; it can also make reader misunderstand what the sentence is trying to accomplish (Gopen, 2004). The other expectation readers have regarding subject and verb is that they will be positioned close to each other: when readers encounter a subject, they start looking for the verb and are not paying much attention to words between subject and verb; thus it is not recommended to place anything of great importance between the subject and its verb (Gopen, 2004; Lebrun, 2011; Day, 1995).

Readers also direct other expectations towards subject position and the beginning of a sentence (Gopen, 2004; Lebrun, 2011). It is a place from where readers expect to find

information to be used as a context for the whole sentence. Readers have two needs they wish to satisfy as soon as possible after they start a sentence: they want to know *whose story* the sentence is, and they want to know *how* the sentence *links backwards* to the previous sentences (Gopen, 2004). In case of one-clause sentences or single clauses, readers interpret "whose story" to be whoever or whatever comes first in subject position; in case of multi-clause sentences the sentence tends to be interpreted as belonging to whoever or whatever comes first in subject position in the sentence's *main clause* (Gopen, 2004). The backward link to the previous sentence is also expected to be found from the beginning of sentence. This place, at the beginning of a sentence, is called the *topic position* (Gopen, 2004; Lebrun, 2011). It holds "old" information that is used in backward linking, and that usually is found from the previous sentence's end. Here, "old" information refers to any piece of material that is familiar to the reader from previous sentences. The length of topic position is not fixed: it continues as long as it is clear that the sentence is beginning; in most cases, this includes the subject but not the verb (Gopen, 2004). In multi-clause sentences, each clause has its own topic position.

Besides the beginning of sentences, readers also have expectations regarding the places of syntactic closure, generally the sentence endings: they expect to find the *most important information* of the sentence from it (Gopen, 2004; Lebrun, 2011). This place at the syntactic closure is called the *stress position*. Whereas in the topic position, the information should be old and familiar, in the stress position it usually is new (Gopen, 2004; Lebrun, 2011). There are few reasons, why it is recommended, and why readers expect the stress position to hold the new and important information. First of all, as mentioned above, readers tend to pay little attention to words between the subject and verb, thus making placing important information between those two structural positions not recommended; on the other hand, stress position, which starts at or after the verb, is not affected by this reader expectation. Secondly, English readers tend to enjoy "delayed gratification", that is, they enjoy the building sense of tension while they read the sentence, and the moment at the end, when the tension breaks, and the "mystery" of that sentence is revealed (Gopen, 2004). Thirdly, English readers have a psychological need for closure and completion; if the sentence does not end with clear and satisfying closure, this psychological need fails to fulfill and may leave reader unsatisfied (Gopen, 2004). These three reasons all relate, according to Gopen (2004), to an old idea of readers emphasizing the importance of endings: *"Aristotle, Cicero, and Quintilian all claim the same for the oratorical Latin sentence. The principle was reiterated*

in the seventeenth century by [...] *compilers of English handbooks, and then again by* [...] *the eighteenth-century Scotsmen* [...]", and *"It has been reconfirmed by research in psycholinguistics, cognitive psychology, and composition theory"*. Gopen (2004) also adds his own interpretation: readers tend to emphasize sentence ends, or moments of syntactic closure, because at that moment they know they can use freely the remainder of their so called *"reader energy"*, which, in turn, produces, to the readers, a sense of emphasis and arrival.

Reader energy is based on idea that readers have and consume mental energy while reading different units of written discourse (Gopen, 2004). For instance, readers consume paragraph energy while reading paragraphs, sentence energy while reading sentences, and clause energy while reading clauses. Reader energy consists of two parts: 1) syntactic energy, and 2) semantic energy. *Syntactic energy* is consumed to clarify the structure of discourse unit (e.g. sentence), and *semantic energy* to clarify the meaning of words (substance) in that unit. These two energy types occur simultaneously while reading, and are zero-sum in terms of their nature: the total of finite available reader energy is divided between these two, so that the more energy is required to accomplish the other, the less is available to the another. For example, if a sentence is structured in a way that is difficult for the reader, reader has to consume majority of his available sentence energy to clarify the structure alone, which leads to insufficient amount of energy left to process the substance of sentence. This, in turn leads to difficulties in comprehending the meaning of that sentence. Gopen (2004) compares this process to breathing: when the reader starts a sentence, or other discourse unit, they take a "breath" and must hold it until the end of that unit when they can release it; the same way the reader regulates his/her reader energy consumption, but only until the end of that discourse unit is in sight. After that point, the reader can freely use the remainder of reserved energy to process the last pieces of information. And that, according to Gopen (2004), is what causes the sense of emphasis and importance for information located at the syntactic closure of a discourse unit.

Progression between sentences

Progression is a process of transforming new information to what is known (Lebrun, 2011); it is about using the information from the previous sentence to contextualize the current one. The previous section, although it also introduced two important terms

used in progression, topic position and stress position, discussed about sentences as isolated units. We now focus on relations and progression between sentences.

Table 2 by Gopen (2004) provides a summary of how sentences are regarded as a part of discourse, with connections to other sentences. There are basically two possible progression schemes (Lebrun, 2011): the *topic based progression*, and *non-topic based progression*. Table 2 describes the former of these. The following sections discuss this matter in following order: first of the non-topic based progression, and then, of the topic based progression.

Table 2: Reader's expectations towards sentence's structure and substance (Gopen, 2004)

Structure	Topic		Stress	Fixed
Substance	Old information ← backward link		New, important information	Movable

Non-topic based progression happens, when there is no explicit topic that links sentences together. Instead, progression is established through one of the following (Lebrun, 2011): 1) through explanation and illustration, 2) through time-based steps, 3) through logical, sequential steps, or 4) through transition words. *Progression through explanation* (point #1) usually happens, when the first of the connected sentences acts like a question (or is one), and the second offers explanation for that question; the question sentence raises need and expectations for an answer, which is then fulfilled in the next sentence, thus establishing progression between sentences. *Progression through illustration* means using visuals to connect sentences. *Time-based progression* (point #2) is used with material that contains chronologically ordered steps, and can be expressed by varying verb tenses (from past to present or from present to future) or with adverbs such as "first", "second" and "finally"; for example, methodology sections usually contain such material. *Logical, sequential progression* (point #3) is established when passage of text contains list of items, that are ordered numerically, or by writer defined order, and that follows implicit or explicit logic (such as cause and effect); for example, this paragraph so far can be considered to have followed this progression type. Finally, point #4 introduces *progression through transition words*, an issue that has already been mentioned with Introduction metrics (Section 3.3 and Listing 25). Transition words are special words ("in addition", "however", ...; see Appendix 1 for more examples) that establish somewhat artificial link between sentences;

according to Lebrun (2011) they are a topic of controversy, and often merely a "convenient way to ignore progression" that should be replaced with implicit progression. Some authors, such as Zeiger (2000) regard transitions as a suitable story-telling technique. Also Lebrun (2011) acknowledges that transition words can be used, when no other progression scheme can be used; for example when connecting two independent sentences together.

Topic based progression uses information at topic and stress positions to establish connection between sentences (Gopen, 2004; Lebrun, 2011). There are three progression schemes, or strategies, that are possible results of filling topic and stress positions according to expectations: 1) topic changing (Gopen, 2004; Lebrun, 2011), 2) topic stringing (Gopen, 2004; Lebrun, 2011), and 3) topic stringing with topic's subclasses (Lebrun, 2011). *Topic changing* mean, that the topic position of the sentence is filled with the information from previous sentence's stress position; this pattern is repeated for successive sentences. Successive sentences, therefore, do not discuss the same topic, nor is it possible to establish lengthy explanations for a single topic. In *topic stringing*, on the other hand, a number of sentences revolve around a constant topic. In topic stringing, the topic position of successive sentences is filled with same information. In topic stringing, therefore, one topic can be expanded and explained in more detail than in topic changing progression scheme. Third topic progression scheme, the *topic stringing with topic's subclasses* is related to topic stringing, with the difference that the exactly same topic is not used in successive sentences. Instead, subclasses, different aspects of paragraph's main topic are used in topic positions in successive sentences to establish connection between sentences.

Expectations towards paragraphs and progression between them

Expectations towards paragraphs are similar to expectations the readers have towards sentences: readers expect certain structural positions be filled with certain substance, and this certain substance to contain material that establishes links between paragraphs (Gopen, 2004). Instead of topic and stress positions, readers expect to find certain substance at *issue* and *point* positions in a paragraph (Gopen, 2004).

The **issue** of a paragraph refers to intellectual boundaries within which the discussion of that paragraph is going to, and should, wander (Gopen, 2004). For example, if an issue is about World War II, the discussion will, and it is expected to, cover different

aspects of the war, like how the 101st Airborne Division participated in D-Day, but not, for example, how the Internet was developed. The issue, in other words, sets the context for further discussion. Readers expect to find it from the beginning (first, or near the first sentence) of a paragraph, at the Issue position (Table 3). They also expect the issue to be developed further during the discussion.

The **point** is the most important idea, within the boundaries set by the issue, that a paragraph contains (Gopen, 2004). It is the mental destination the reader is wanted to arrive. Most of the time, readers prefer being explicitly, and in a single sentence, told what the point is (Gopen, 2004). They also expect the point to be found at a Point position (Gopen, 2004). The Point position is either at the last sentence of issue, just before the discussion begins, or at the last sentence of discussion, near the ending of a paragraph (Table 3). Which of these places is preferred by readers, depends on the type of paragraph: for the first and last paragraphs of a section or the whole document, the point is expected to come after the discussion; for the most medial paragraphs, it is the opposite: the point is preferred to come up front. Gopen (2004) explains that at the first paragraph readers are unfamiliar with the issue, and need the context to set up before reading about the point; at the last paragraph, on the other hand, the point in the end brings a satisfactory end to the whole discourse, and is therefore preferred. The preference for having the point up front at the medial paragraphs originates from the non-linear nature of scientific reading: readers tend to read the first and final paragraphs, but skip the paragraphs between, if they do not immediately find something from the paragraph that motivates them to read further.

Table 3: Reader's expectations towards paragraphs. Readers expect the paragraph to start with issue and either provide the point before the discussion, or after it. (Gopen, 2004)

Structure	Issue	<Point$_1$>	Discussion	<Point$_2$>	Fixed
Substance	issue	point$_1$		point$_2$	Movable
	First sentence			Last sentence	

⟵ Paragraph ⟶

The progressions between paragraphs are similar to those between sentences; instead of topic and stress, readers look for material at the Issue and Point positions to establish connection (Gopen, 2004). A progression between paragraphs can be established by having material 1) at the end of a paragraph that links forwards to the next paragraph,

or 2) at the beginning of a paragraph that links backwards to the previous paragraph (Gopen, 2004). The former case informs reader at the end of the current paragraph what is going to happen next, before moving on to the next one. The latter case, on the other hand, is similar to the progression between sentences, and its topic–stress paradigm: in this case, the material at the Issue position can be used to link backwards either to a) material at the previous paragraph's Issue position, which can contain issue and point, or to b) material at the end of previous paragraph, which often contains the point (Gopen, 2004).

Fluidity metric algorithm

We are now ready to describe the algorithm used to evaluate fluidity in the current implementation of SWAN. As a memory refreshment, Table 4 summarizes the meaning of topic and stress, and adds some new terms: *strong topic*, *weak topic*, *strong stress*, and *weak stress*. These terms are an expansion by Lebrun (2011) to the original topic and stress by Gopen (2004), and are used in the fluidity algorithm. The fluidity metrics the algorithm follows, are developed by Lebrun (2011). These metrics focus on progression between sentences, and do not evaluate e.g. whether the subject and its verb are at suitable distance from each other, or whether any kind of progression between paragraphs exist. The algorithm, in a form of pseudocode, is based on the current implementation of SWAN. For the sake of clarity, some less essential parts have been simplified, or completely omitted. The following paragraphs present major steps used in the algorithm, as well as corresponding listings of pseudocode. The complete pseudocode for the algorithm is provided in Appendix 3B.

Table 4: Definitions for terms used in the fluidity metric algorithm (Gopen, 2004)

Traditional terms, by Gopen (2004)	
Topic	Old information that links backwards, found at the beginning of sentence
Stress	New, important information, at the syntactic closure

Expansion for topic and stress, by Lebrun (2011)	
Strong topic	Noun found from sentence's main clauses
Weak topic	Noun or verb-derived noun found from elsewhere in the sentence

Strong stress	Word suitable for stress, meeting one, or more of the following criteria: 1. Word is a noun appearing before the first punctuation mark 2. Word is a verb-derived noun, derived from a verb from the main sub-clause 3. Word is a noun appearing after the last punctuation mark or last conjugated verb 4. Word is a noun of the main clause appearing after the conjugated verb, and the main clause contains a topic 5. Word is a noun preceded by a number
Weak stress	Word otherwise suitable for stress, but not meeting criteria for strong stress

The following list describes the major steps used in the algorithm. The same major steps, and how they flow, can be seen in visual form in Figure 4. These steps will be discussed in the following paragraphs.

1. Pre-process the inputted text.
    1.1. Remove literature references from paragraphs.
    1.2. Split text into paragraphs.
    1.3. Split paragraphs into sentences and label words with part-of-speech tags.
    1.4. Remove “short stubs” from sentences.
2. Take each paragraph under evaluation independently from the other paragraphs (fluidity is not evaluated between paragraphs).
3. Process sentences from paragraph one by one until the last sentence of that paragraph is processed. Then repeat the process with the next paragraph.
    3.1. If sentence under evaluation is the first sentence of the paragraph, define default word sets and move to next sentence.
    3.2. If not the first sentence, flag for potential placebo transitions for sentence (see Appendix 3 for the full list).
    3.3. If not the first sentence, check if the sentence begins with words that indicate fluidity.
        3.3.1. If fluid words are found, define default word sets and move to next sentence.

3.3.2. Otherwise, check progression of sentence S_n in relation to previous sentence, S_{n-1}.

3.3.2.1. If progression with S_{n-1} is found, move to the next sentence.

3.3.2.2. If no progression is found with S_{n-1}, repeat the process with S_{n-2}, and if still no progression is found, with S_{n-3}.

3.3.2.3. Move to the next sentence and repeat the process.

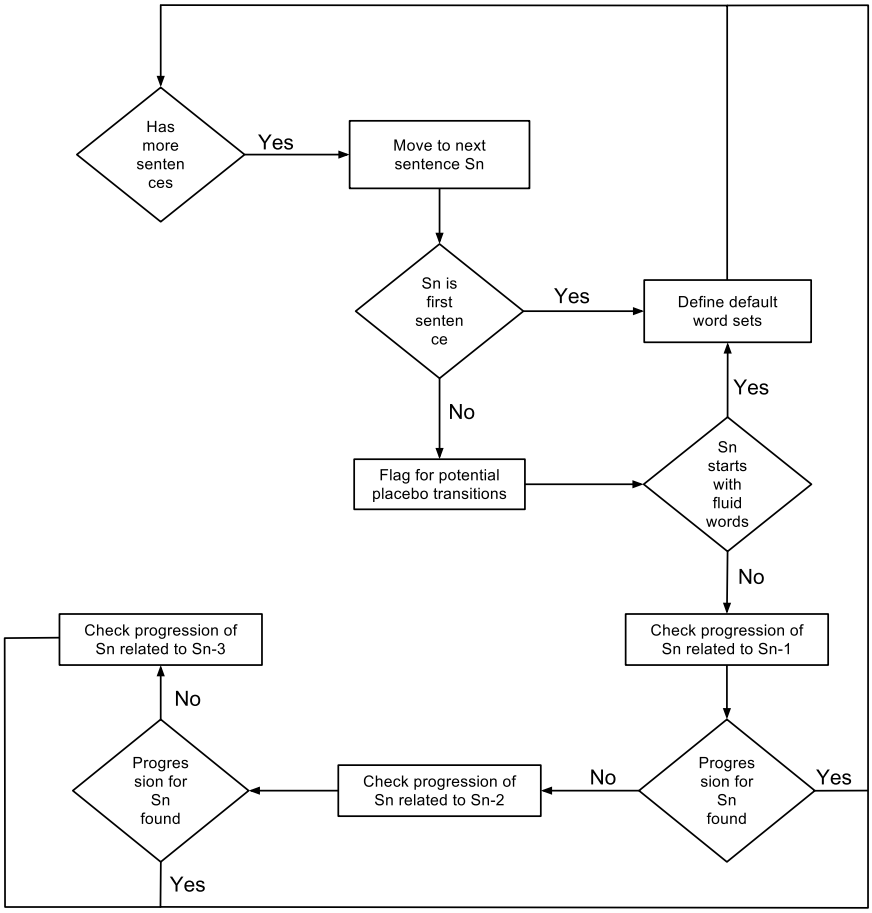

Figure 4: An overview of fluidity metric algorithm

Step 1. First step of the algorithm is to prepare inputted text for evaluation. This preparation includes removing literature references (e.g. "Smith (2000)", "(Smith, 2000)", "[1]", and "[1, 2, 3]"), splitting the text into paragraphs, and paragraphs into sentences, tagging the words (giving part-of-speech categories), and removing short stubs (short expressions that start with "it" or "there" and end with "that"; e.g. "It is obvious that ..."). The pseudocode for this is described in Appendix 3B, in Listing 39 (line 25). The purpose of this preparation is to process text ready for the algorithm, and also to remove any material that could potentially cause disruptions during evaluation. Example

1 provides an example input, and output for this step. In this example, there was only one paragraph, and no short stubs, but otherwise the text would have been split into paragraphs, and the short stubs would have been removed.

Example 1. Preprocessing

Input

> One of the biggest challenges in automatic speaker recognition is obtaining invariance across varying operating conditions, while retaining maximum speaker variability. Different handset type, transmission line/coding, and background noise are typical factors, which lead to signal mismatch across training and recognition. For a speaker recognition system to be useful in practice it needs to be optimized against the mismatch problem. Various approaches have been proposed for tackling the invariance problem, including robust feature extraction (Mammone et al., 1996), feature normalization (Pelecanos and Sridharan, 2001), model transformation (Kenny et al., 2007; Teunen et al., 2002; Vogt and Sridharan, 2008), and match score normalization (Auckenthaler et al., 2000; Reynolds et al., 2000).[1]

Output

> **#1** One of the biggest challenges in automatic speaker recognition is obtaining invariance across varying operating conditions, while retaining maximum speaker variability. **#2** Different handset type, transmission line/coding, and background noise are typical factors, which lead to signal mismatch across training and recognition. **#3** For a speaker recognition system to be useful in practice it needs to be optimized against the mismatch problem. **#4** Various approaches have been proposed for tackling the invariance problem, including robust feature extraction, feature normalization, model transformation, and match score normalization.
>
> Sentences are also part-of-speech tagged (see Section 4.3). For instance, sentence #1 would look like this:
> One/CD of/IN the/DT biggest/JJS challenges/NNS in/IN automatic/JJ speaker/NN recognition/NN is/VBZ obtaining/VBG invariance/NN across/IN varying/VBG operating/VBG conditions/NNS ,/, while/IN retaining/VBG maximum/JJ speaker/NN variability/RB ./.

Steps 2 – 3. After preprocessing, the actual evaluation starts. The evaluation focuses

[1]This example, and the ones following in this section, use text from the following paper: V. Hautamäki, T. Kinnunen, P. Fränti (2008). Text-independent speaker recognition using graph matching. *Pattern Recognition Letters*, 29(9):1427-1432.

on one sentence at a time. It aims in three things: 1) in detecting the topic of a sentence, 2) in detecting the stress of a sentence, and 3) in detecting the progression type of a sentence. Topic is mainly detected by matching words (nouns, or in certain situations, verb-derived nouns) from the sentence under evaluation with the previous sentence(s). Stress words are suitable words that have not been detected as topic words. Topic and stress words together are referred in the algorithm as the sentence's *wordset*. The progression type of a sentence depends on how, if at all, a topic is found; whether it is found from a topic position and from the immediately preceding sentence or further on. There are, however, a few situations when it is not necessary to use previous sentence(s) as help: the first of these situations is when the first sentence of a paragraph is evaluated (the other situation will be discussed in the following paragraphs). This is because each paragraph is treated as an isolated unit, and therefore the first sentence is not required to be linked with the last sentence of previous paragraph. This means, that the first sentence in a paragraph is given a default wordset and no particular progression type. Example 2 describes one such case. The default wordset is defined so that the sentence topic is set to be the subjects from the sentence's main clause, and the stress to be all other nouns and verb-derived nouns; the first sentence acts as the basis for further topic matching. The detailed pseudocode is given in Appendix 3B, in Listing 39 (lines 2 and 59).

Example 2. Processing the first sentence. The sentence is given the defaults wordset: main clause's subjects are set as topic words, and the other noun and verb-derived nouns as stress words.

Input

> **#1** One of the biggest challenges in automatic speaker recognition is obtaining invariance across varying operating conditions, while retaining maximum speaker variability.

Output

> **#1 One** of the biggest *challenges* in *automatic speaker recognition* is *obtaining* **invariance** across *varying operating conditions*, while *retaining* maximum *speaker variability*.
>
> The main clause of the sentence is underlined.
> **Topic words**: [one, invariance] (also subjects of the sentence's main clause)
> *Stress words*: [challenges, speaker, recognition, obtaining, varying, operating,

conditions, retaining, speaker, variability]
Progression type: Not specified

Steps 3.2. – 3.3. Sentences following paragraph's first sentence are mainly evaluated in relation to previous sentence(s). The evaluation is therefore more complex and contains more steps (see Appendix 3B, Listing 39, and line 5 for pseudocode). The first step is to check the sentence for the so called *placebo transitions* (Step 3.2.). Placebo transitions are words such as "additionally", "furthermore" and "however" (see Appendix 3A for the full list) that begin sentences and establish often artificial connection; should these occur and the sentence is not otherwise proven to be fluid, a warning is given to the user. Then, regardless of placebo transitions, the evaluation continues to check the actual progression type for the sentence. This contains two possible cases: 1) sentence is proven to be fluid via *fluidity words* (the other situation mentioned in the previous paragraph), or 2) attempted to be proven fluid by using *non-deterministic topic search* with previous sentence(s). First case contains simple word matching by using predefined set of words that establish fluidity (fluid words and pronouns; see Appendix 3A); the pseudocode for it is described in Appendix 3B, in Listing 39 (line 66). The latter case, using non-deterministic algorithm for topic search, is more complicated, and will be discussed in the following paragraph.

Step 3.3.2. The non-deterministic topic search primarily aims in finding progression with the sentence S_n that is under evaluation and the one immediately preceding it, S_{n-1}. However, that is not always possible, and therefore the search will secondarily try to find progression from sentences down to S_{n-3}. As a result of these searches, the sentence under evaluation has a defined wordset and a progression type, which can be (in order of decreasing fluidity) either *fluid*, *inverted topic*, *out of sync* or *disconnected*. The search consists of two checkups: 1) one directed to the sentence's main clauses only, and 2) one directed to the full sentence. The first of them, the main clause checkup, starts the checking, and aims in finding the strong topic for the sentence; in other words, it tries to match words at sentence's main clause with words in previous sentence's topic and stress positions (see Appendix 3B, Listing 39, and line 79 for pseudocode). The full sentence checkup, on the other hand, uses all nouns and verb-derived nouns from the sentence in order to detect its (weak) topic, stress and progression type (see Appendix 3B, Listing 39, and line 89 for pseudocode). As mentioned earlier, the primary goal is to find progression with the immediately preceding sentence S_{n-1}, but the algorithm will continue searching connections, if necessary, with S_{n-2}, and if still

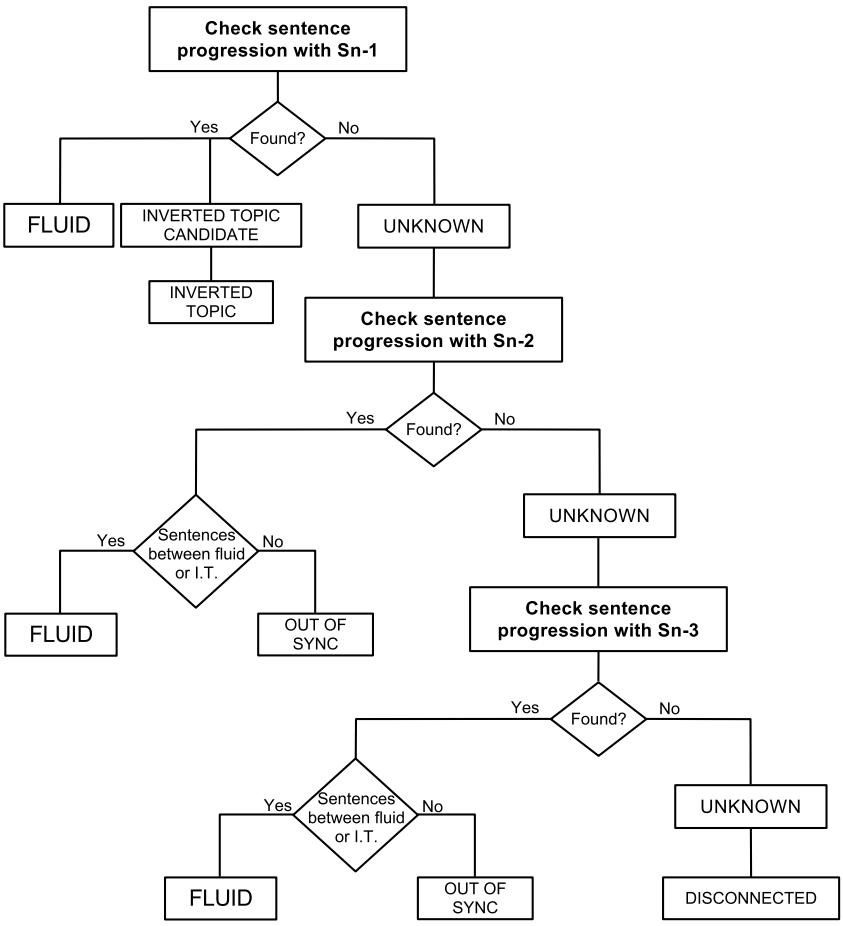

Figure 5: Inputs and results for different check rounds in sentence progression evaluation.

no progression, with S_{n-3}. The check rounds and their results differ from each other. Figure 5 shows the outcomes of each check round. Next, we will go through each of these rounds.

The first check round, as all check rounds, starts with checking sentence's main clauses. If a strong topic is found, the sentence is *fluid*, and rest of the algorithm only searches and defines the wordset. If a strong topic is not found, the topic search continues in the second part of algorithm by checking the whole sentence. The algorithm starts with considering only nouns to be suitable for topic. Word by word it tries to match nouns from S_n with topic and stress words from S_{n-1}. If a match is found, it is marked as weak topic, and the sentence is *fluid* (see Example 3). After this point, also verb-derived nouns are considered suitable to be topic words, and the rest of the algorithm only searches and defines rest of the wordset; topic words are searched only until the first conjugated verb, after which, nouns, or verb-derived nouns that match with topic

and stress words from S_{n-1}, are considered stress words.

Example 3. Processing the sentences after the first sentence. Fluid sentence #3 with #2 (S_{n-1}).

Input

> [...] **#2** Different handset *type*, transmission line/coding, and background noise are typical factors, which lead to signal *mismatch* across *training* and **recognition**. **#3** For a speaker recognition system to be useful in practice it needs to be optimized against the mismatch problem.

Output

> [...] **#2** Different handset *type*, transmission line/coding, and background noise are typical factors, which lead to signal *mismatch* across *training* and **recognition**. **#3** For a *speaker* **recognition** *system* to be useful in *practice* *it* needs to be optimized against the *mismatch problem*.
>
> The main clause of the sentence is underlined.
> **Topic words**: [recognition]
> *Stress words*: [speaker, system, practice, it, mismatch, problem]
> Progression type: Fluid; matching word with S_{n-1} recognition

However, if no topic is found before the first conjugated verb is reached, the algorithm has failed to detect fluid progression between sentences S_n and S_{n-1}. If, however, the topic is found after the conjugated verb, the sentence is marked as a *candidate for inverted topic*. After the rest of the sentence is processed, the algorithm looks for the results. If the sentence was marked as fluid, the algorithm continues to the next sentence. If the sentence was marked as a candidate for inverted topic, the progression between this, and the previous sentence is marked as *inverted topic* (see Example 4), and the algorithm continues to the next sentence. If no topic was found, the sentence is marked temporarily as *unknown*, and an additional check round, this time with S_{n-2}, is needed.

Example 4. Processing the sentences after the first sentence. Inverted topic sentence #2.

Input

> **#1 One** of the biggest *challenges* in *automatic speaker recognition* is *obtaining* **invariance** across *varying operating conditions*, while *retaining* maximum *speaker*

variability. **#2** Different handset type, transmission line/coding, and background noise are typical factors, which lead to signal mismatch across training and recognition.

Output

#1 One of the biggest *challenges* in *automatic speaker recognition* is *obtaining* **invariance** across *varying operating conditions*, while *retaining* maximum *speaker variability*. **#2** Different handset *type*, transmission line/coding, and background noise are typical factors, which lead to signal *mismatch* across *training* and **recognition**.

The main clause of the sentence is underlined.
Topic words: [recognition]
Stress words: [type, mismatch, training]
Progression type: Inverted topic; matching word recognition

The second and third check rounds are almost identical with the first one. The algorithm checks first sentence's main clauses, and after this the full sentence; this time by trying to match topic and stress words from S_{n-2}, if it is the second round, or from S_{n-3}, if it is the third round. If a topic is found from the main clause checkup, or before the first conjugated verb from the full sentence checkup, instead of immediately marking the sentence fluid, an additional checkup will be performed. This checkup looks for the progression types of sentences S_{n-1}, S_{n-2}, and in case of third round, also S_{n-3}. If all of the sentences are either fluid or inverted topic, sentence S_n is marked *fluid* (see Example 5), otherwise, *out of sync*. If a topic is found *after* the first conjugated verb, or if no topic is found at all, and it is the second check round, algorithm proceeds to the third check round. If it is already the third check round, no additional check rounds are performed, and the sentence is marked as *unknown*.

Example 5. Processing the sentences after the first sentence. Fluid sentence #8 with #5 (S_{n-3}).

Input

[...] **#5** State-of-the-art text-independent speaker *recognizers* use mean *subtraction* at the *utterance level*, often referred to as cepstral mean subtraction (*CMS*) in the *context* of cepstral **features**. **#6** The *assumption* in mean **subtraction** is that all the *feature vectors* have been translated by an unknown channel-dependent

vector. **#7** By **subtracting** the *mean* from both the *training* and *testing* **vectors**, the matching is less affected by this *bias*. **#8** For clean data (no channel mismatch), CMS degrades accuracy.

Output

[...] **#5** State-of-the-art text-independent speaker *recognizers* use mean *subtraction* at the *utterance level*, often referred to as cepstral mean subtraction (*CMS*) in the *context* of cepstral **features**. **#6** The *assumption* in mean **subtraction** is that all the *feature vectors* have been translated by an unknown channel-dependent *vector*. **#7** By **subtracting** the *mean* from both the *training* and *testing* **vectors**, the matching is less affected by this *bias*. **#8** For clean *data* (no *channel mismatch*), **CMS** degrades *accuracy*.

The main clause of the sentence is underlined.
Topic words: [CMS]
Stress words: [data, channel, mismatch, accuracy]
Progression type: Fluid; matching word with S_{n-3} (#5) CMS; Sentences #6 and #7 determined fluid previously.

After the progression type is defined, or all possible check rounds are went through, the algorithm does the final definitions for the sentence, before proceeding to evaluate the next one. During the check rounds, the algorithm stored wordsets for each check round. If the sentence was marked as fluid or out of sync, the final wordset (the one that will be used by the next sentence) is taken from the latest check round. If the sentence was marked as inverted topic, instead of the last, the wordset from the first check round is used. If the algorithm failed to detect progression with any of the previous sentences (progression type was marked as unknown), the sentence is considered to contain a new topic that has nothing in common with the previous ones; thus a default wordset is defined for the sentence and the sentence is given the final progression type, *disconnected* (see Example 6). After this, the evaluation procedure is repeated with the next sentence. When all sentences for all paragraphs are evaluated, the results are given to the user. If a sentence was detected to contain placebo transitions, and not being fluid, a warning is given. The problematic sentences, and topic and stress words, are highlighted.

Example 6. Processing the sentences after the first sentence. Disconnected sentence #9.

Input

[...] **#6** The *assumption* in mean **subtraction** is that all the *feature vectors* have been translated by an unknown channel-dependent *vector*. **#7** By **subtracting** the *mean* from both the *training* and *testing* **vectors**, the matching is less affected by this *bias*. **#8** For clean *data* (no *channel mismatch*), **CMS** degrades *accuracy*. **#9** A general affine channel/environment model includes rotation and scaling of the feature vectors in addition to the additive bias.

Output

[...] **#6** The *assumption* in mean **subtraction** is that all the *feature vectors* have been translated by an unknown channel-dependent *vector*. **#7** By **subtracting** the *mean* from both the *training* and *testing* **vectors**, the matching is less affected by this *bias*. **#8** For clean *data* (no *channel mismatch*), **CMS** degrades *accuracy*. **#9** A general *affine channel/environment* **model** includes **rotation** and **scaling** of the *feature vectors* in *addition* to the additive *bias*.

The main clause of the sentence is underlined.
Topic words: [model, rotation, scaling]
Stress words: [affine, channel/environment, feature, vectors, addition, bias]
Progression type: Disconnected. No topic word in sentence #9 match topic and stress words in sentences #6–#8.

4 Java implementation of Scientific Writing Assistant

Scientific Writing Assistant (SWAN) is a project, that has been under active development since 2009 at the School of Computing, University of Eastern Finland (see Figure 6 for project timeline). The development group[2] has consisted of university staff members to manage the project, and university students (both Master's and PhD students), who participate in development for a certain period of time. The original idea, as well as the underlying evaluator metrics, have been developed by Mr Jean-Luc Lebrun[3], who is an independent scientific writing trainer. The project's purpose has been to produce a computer-assisted tool that can be used to improve the readability of scientific manuscripts.

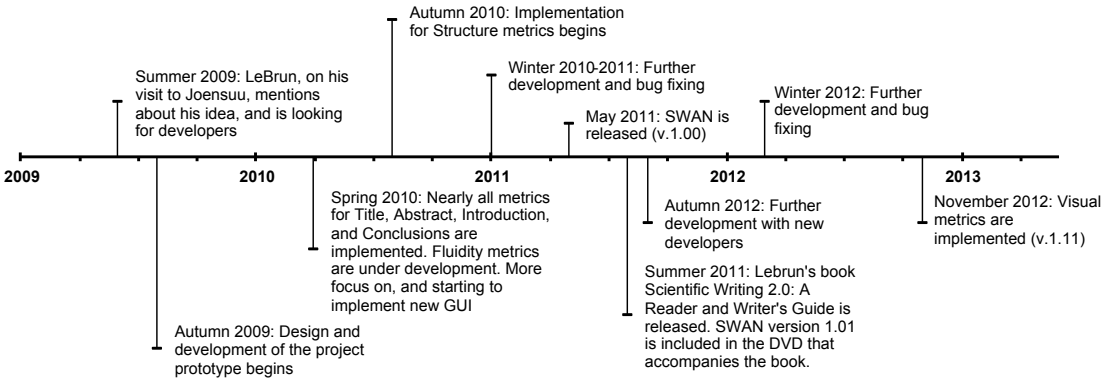

Figure 6: Timeline of the SWAN project.

The current implementation (2009–2013) of Scientific Writing Assistant is developed with Java programming language, and runs in Java Runtime Environment (JRE)[4] version 1.6 and newer. SWAN is a multiplatform application, that supports operating systems that can run JRE. These include Microsoft Windows, Apple's Mac OS X, and Linux distributions, such as Ubuntu, that contain graphical desktop. Requirements for running SWAN are:

- 512 MB or more RAM
- Java Runtime Environment 1.6 or newer
- For 32-bit Operating Systems a 32-bit JRE must be installed; 64-bit Operating Systems may run both 32-bit and 64-bit JREs

[2] `http://cs.uef.fi/swan/members.html`
[3] `http://www.scientific-writing.com/`
[4] `http://www.java.com/en/download/faq/whatis_java.xml`

SWAN consists of two kinds of evaluation metrics: manual and automatic. *Manual* evaluation is a self-test way of getting feedback from a text, and depends entirely on user-interaction. Currently, SWAN contains only one manual evaluation task: manual fluidity evaluation. It is related to the automatic fluidity evaluation, and is aimed for writers who want to test semantic progression for their text. *Automatic* evaluation, in turn, refers to evaluation which is, after some initial user-interaction, done by the computer. All of the metrics described in Section 3 are carried out automatically by SWAN. Automatic evaluation consists of custom code that implements the quality metrics, and Natural Language Processing (NLP) tools from The Stanford Natural Language Processing Group[5], that supports the custom code.

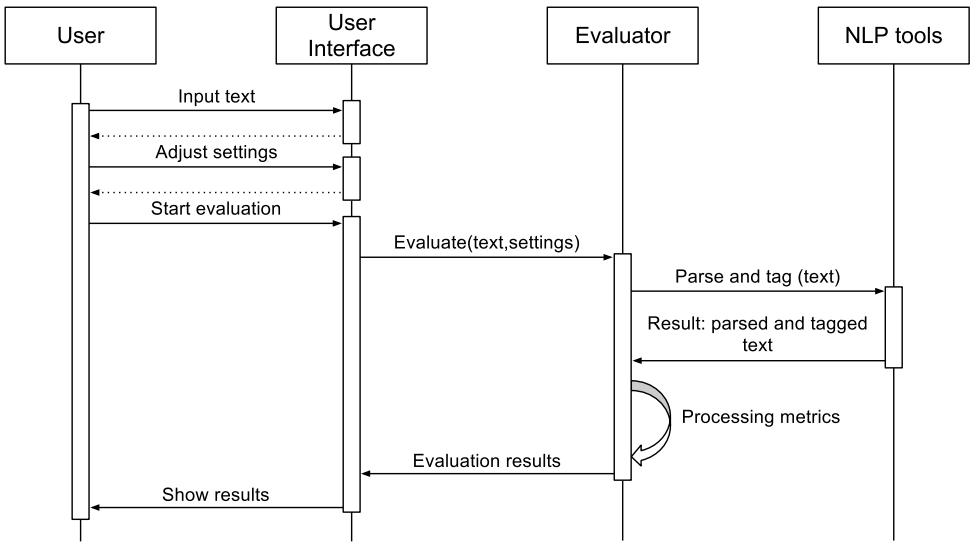

Figure 7: An example of basic use flow for SWAN.

The basic use flow for SWAN contains the following major steps: 1) inputting the text, 2) adjusting settings, 3) starting the evaluation, and 4) viewing the results (Figure 7). For inputting, there are three alternatives: either the user manually copy and pastes text from their source document to SWAN, the user imports their document by using SWAN's semi-manual document structure parser, or the user reloads a previously inputted text from a SWAN save file. Adjusting settings typically includes light manual work in preparation for evaluation. For example, user is asked to join keywords from the title and select, which reflect contribution, or highlight sections from the abstract. Evaluation is automated; it contains rule-based checkings according to the metrics. The result for evaluation is a list of both potential suggestions for improvements, and

[5] `http://www-nlp.stanford.edu/`

positive mentions, where the text has met the criteria set by the metrics. The results, as well the inputted source document and the settings adjustments can be saved to the user's computer.

SWAN employs a set of external libraries. These libraries are listed in Table 5. These libraries are used, for instance, for natural language processing, importing and exporting documents, and setting the appearance for the graphical user interface.

Table 5: External libraries used by SWAN. All URLs valid 25.5.2013.

Component	**Usage**	**Component webpage**
Stanford Parser	Natural language processing	`http://nlp.stanford.edu/software/lex-parser.shtml`
Stanford POS Tagger	Natural language processing	`http://nlp.stanford.edu/software/tagger.shtml`
SnowBallStemmer	Extracting word stems	`http://snowball.tartarus.org`
Apache Tika	Extracting documents	`http://tika.apache.org`
Substance	Look and feel for the graphical user interface	`http://insubstantial.github.io/insubstantial/substance`
Trident	Animation library used by Substance	`https://kenai.com/projects/trident`
Apache Commons Math	Mathematical processing (calculations)	`http://commons.apache.org/proper/commons-math`
JFreeChart	Generating graphs	`http://www.jfree.org/jfreechart`
JCommon	Used by JFreeChart	`http://www.jfree.org/jcommon`
XStream	Serializing Java objects to XML and back	`http://xstream.codehaus.org`

4.1 Automatic Evaluation

The automatic evaluation consists of seven parts: title, abstract, introduction, conclusions, structure, fluidity, and visuals. Each of these evaluations focuses on different parts or aspects of a scientific paper. First four (title, abstract, introduction and conclusions) assess the text quality of theirs respective sections. Structure evaluation focuses on paper's outline: headings and sections underneath them. Fluidity evaluation assesses text progression, while visuals evaluation focuses on visuals (figures and tables) found from the paper. Each of these evaluations are based on the quality metrics described in Section 3.

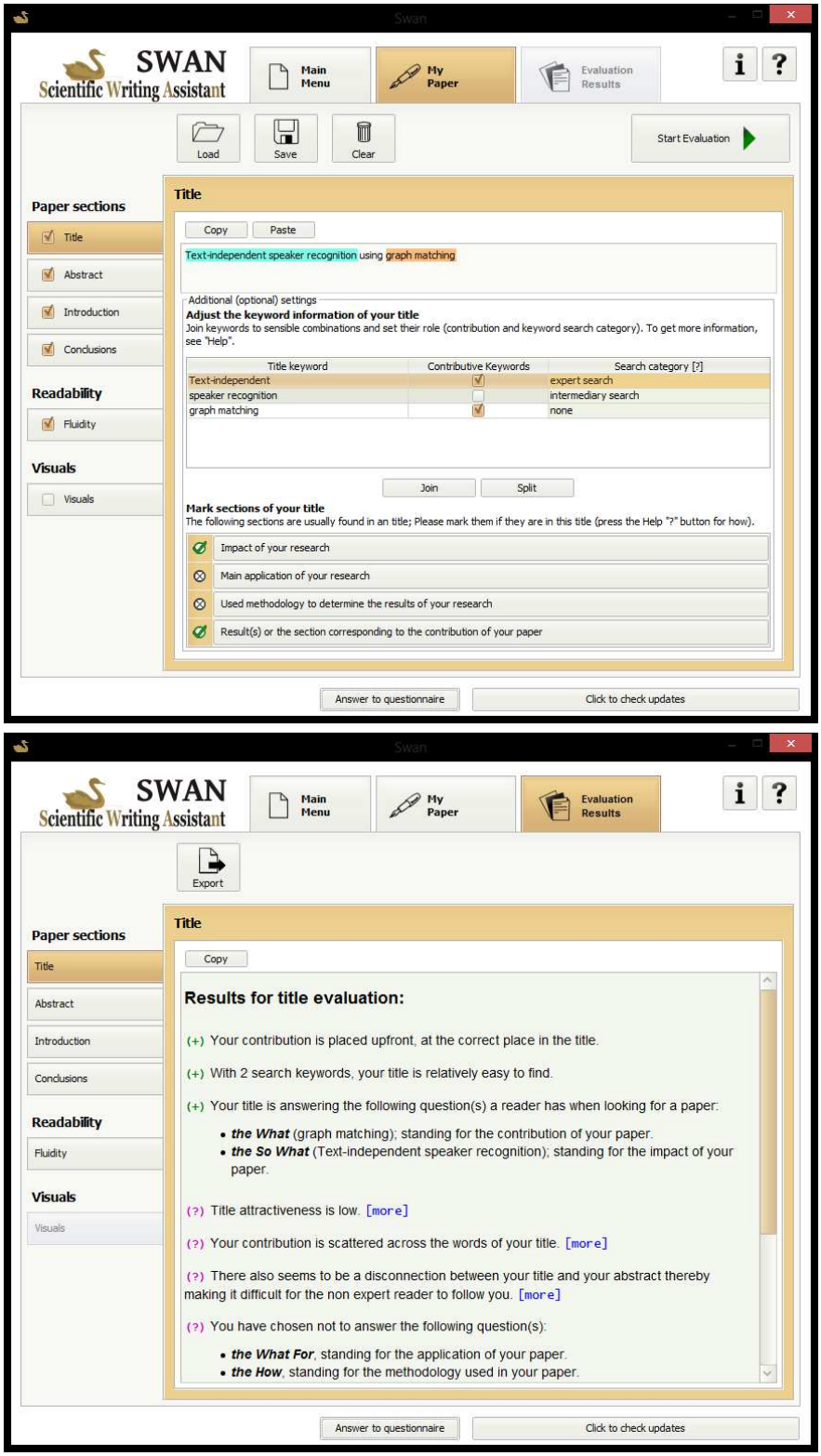

Figure 8: An example of automatic evaluation. Text can be either insert manually, or with document import. After some settings (the upper image), the evaluation can begin. After the program has processed through metrics, the results are shown (the lower image)

4.2 Manual Evaluation

Manual fluidity evaluation is currently the only manual evaluation task in SWAN. It is meant for writers who want to evaluate the semantic progression for a given text. It is based on the same idea as the automatic fluidity evaluation (see Section 3.6), although it has been simplified to suit better as a manual task. Instead of asking the users to identify topic and stress from sentences, users are only required to read a sentence from the text they have inputted, and answer whether they can anticipate the topic of the succeeding sentence based only on this given sentence. The manual fluidity evaluation requires substantially more effort from the user than its automatic equivalent, but, as the automatic evaluation is still a work in progress, and it e.g. does not currently work on semantic level of the text, manual evaluation's accuracy is also greater, and it works in situations, in which the automatic evaluation fails. Such situation is described in Example 7. The automatic evaluation is unable to connect the sentences, because they do not share common words, and because the second sentence does not begin with fluid words. A human evaluator, on the other hand, is able to see that "lions" mentioned in the first sentence, are related to the "distant relative" and "domesticated cat" mentioned in the second sentence: both a lion and a cat belong to the feline species, and are thus relatives. The basic algorithm for manual fluidity is given in Figure 10.

Example 7. An example of situation in which the manual fluidity performs better than the automatic fluidity.

Input for automatic evaluation

> **[#1]** Lions hunt large game, such as antilopes. **[#2]** The distant relative, the domesticated cat, on the other hand, has to settle for little mice.

Output for automatic evaluation

> **[#1]** *Lions* hunt large *game*, such as *antilopes*. **[#2]** The distant **relative**, the *domesticated cat*, on the other hand, has to *settle* for little *mice*.
>
> **Topic words**: [relative]
> *Stress words*: [domesticated, cat, settle, mice]
> Progression type: Disconnected

The basic idea for the evaluation is to go through, one by one, all the sentences in a text. Only the sentence that has the focus, is shown fully to the user. The other sentences

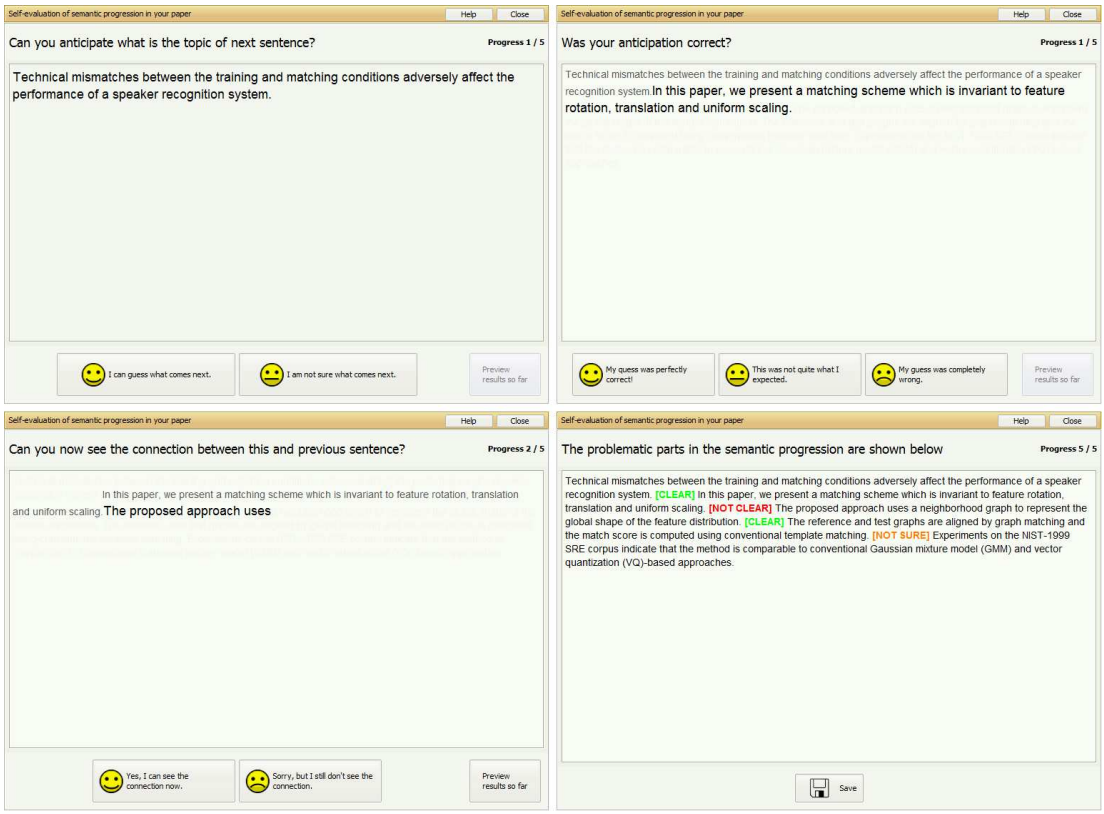

Figure 9: An example of manual fluidity evaluation[6]. The upper-left image shows the situation at the start of each step. After the user answered that they can anticipate the topic, the evaluation asks whether this anticipation was correct (the upper-right image). If at some point the user cannot anticipate the topic, the beginning of the next sentence is shown (the lower-left image). After all sentences has been processed, the results are shown (the lower-right image).

are hidden. Based on this focus sentence, the user is asked whether they can anticipate the topic of the next sentence. If the user thinks they can anticipate the topic, the next sentence is shown to them, and the user is asked how correct their anticipation was: a) perfectly correct, b) not quite the expectation, or c) completely wrong. According to this answer, a progression category of *fluid*, *not sure*, or *disconnected* is given to the connection between the sentences. Progression category "not sure" refers to situations, when the user managed to anticipate the topic somewhat, but it was not entirely as they expected, and so the writer should consider whether they need to change either of the sentences. After this, the current focus sentence is hidden, and the sentence of which topic the user was asked to anticipate, becomes the focus sentence. Then, the

[6]The example uses the abstract from the following paper: V. Hautamäki, T. Kinnunen, P. Fränti (2008). Text-independent speaker recognition using graph matching. *Pattern Recognition Letters*, 29(9):1427-1432. The results of this example are an outcome of demonstration of this tool, and not actual results of any evaluation.

anticipation of the topic of the next sentence based on this new focus sentence is asked. If the user is not sure about the next sentence's topic, the evaluation shows them a part of the next sentence: the beginning of a sentence up to the first verb. The user is asked again for the anticipation. If they are still not sure about the topic, the evaluation shows the full sentence, marks connection between sentences "disconnected", and moves on to the next one (respectively, if the user was, after seeing the beginning of the next sentence, sure about the topic, sentences are given progression category of "fluid"). The evaluation repeats these steps for every sentence, until the second to last is given focus, and the topic for the last sentence is asked. After this, the results are shown.

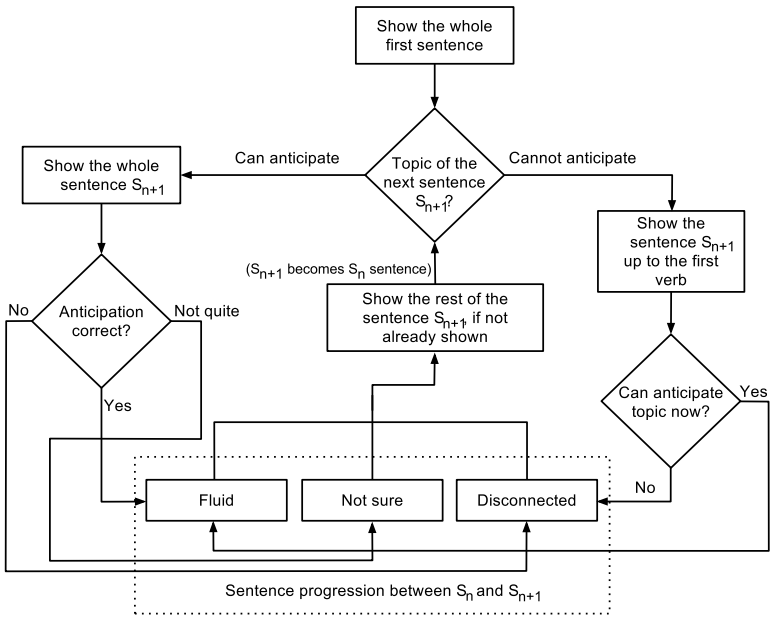

Figure 10: Flowchart

4.3 Tools for Natural Language Processing

To support the evaluation metrics, SWAN uses natural language processing (NLP) tools, namely the Stanford Parser and Stanford Part-Of-Speech (POS) Tagger libraries. The Stanford Parser library is a collection of probabilistic natural language parsers, such as optimized *Probabilistic Context Free Grammar* (PCFG) and lexicalized dependency parsers, that parse the grammatical structure of a given text (Klein and Manning, 2003). The Stanford POS Tagger is a library of tools that label words in a given sentence with part-of-speech tags (Toutanova et al., 2003). The tagger labels POS tags according to the tagset in Penn Treebank. The Penn Treebank is a large corpus of

English words, that contains tagset for part-of-speech labeling (Marcus et al., 1993). The performance tests by Klein and Manning (2002, 2003) have shown an accuracy between 87 % to over 91 % for dependency parsing with the Stanford Parser, and an accuracy of approximately 97 % for part-of-speech tagging with the Stanford POS Tagger on the Penn Treebank WSJ (Toutanova et al., 2003).

Part-Of-Speech Tagging

The Stanford POS Tagger is used to detect word classes (part-of-speech) from the words in the text. This detection divides words to major word classes (nouns, adjectives, verbs, prepositions, etc) and these major classes into subcategories (e.g. nouns into proper, singular or plural nouns, and verbs into different tenses). The tagged text is a basis for nearly all the evaluations in SWAN; it is used, for instance, to detect keywords from the title, calculate amount of different verb tenses, or punctuation marks, or to check if a particular word belongs to a certain word class. An example of outcome for detecting part-of-speech for a sentence is shown in Listing 36. Each word has been associated with its Penn Treebank tag. With this output, a metric could e.g. check whether the given sentence contains an adjective (and in this case the result would be positive: POS tag `JJ` refers to adjectives).

```
Input sentence:
The brown dog chases a black cat.

Tagged result:
The/DT, brown/JJ, dog/NN, chases/VBZ, a/DT, black/JJ, cat/NN, ./.
```

Listing 36: Tagging a sentence with Stanford POS Tagger. The POS tags in this example: DT = determiner; JJ = adjective; NN = noun, singular or mass; VBZ = verb, 3rd ps. sing. present

Parsing and detecting dependencies

The Stanford Parser is currently mainly used in SWAN for extracting grammatical structure from sentences, splitting a given text into sentences, and stemming words. Extraction of grammatical structure gives information both from the roles of individual words in a sentence (e.g. which words are subjects or objects), and the relations be-

tween them (Klein and Manning, 2003); these relations are called *typed dependencies* (De Marneffe and Manning, 2008a,b). The parser bases its Stanford typed dependencies definitions on the same Penn Treebank tagsets, as the POS Tagger (De Marneffe and Manning, 2008a). The parser also uses this information in splitting text into sentences with its `WordToSentenceProcessor`[7]. Word *stemming* means computing the base form of words by removing inflections from the word. The Stanford stemming tool, called `Morphology`, is based on the works of Minnen et al. (2001). In SWAN, stemmed words are used in comparing words together, to allow words with different inflections to be matched. For increased accuracy in these comparisons, SWAN uses also another stemmer, the SnowBallStemmer[8], which is based on the classical *Porter stemmer algorithm* (Porter, 1980). In case of a failure to match words stemmed with the Morphology, the word comparator, as a fallback, stems words with the SnowBallStemmer, and performs second comparison. An example of this is given in Listing 37. The result for Morphology is two different words, which can not be compared directly (although, in this case a suitable regular expression would be able to match the words). The result for the SnowBallStemmer, on the other hand, is two identical strings, which can be directly compared.

```
Input words:
  simulated
  simulation

Result with the Morphology:
  simulated  => simulated
  simulation => simulate

Result with the SnowBallStemmer:
  simulated  => simul
  simulation => simul
```

Listing 37: Stemming words with the Morphology and the SnowBallStemmer. The example shows differences between the results of the two stemmers.

Table 6 shows two examples of outcomes for parsing a sentence with the Stanford Parser. The parser extracts sentence's grammatical structure (parsed sentence), and detects typed dependencies between the words in the sentence (typed dependencies).

[7] See the documentation from `http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/process/WordToSentenceProcessor.html`

[8] `http://snowball.tartarus.org/`

```
Example A (active voice
    sentence):
Input sentence:
  The brown dog chases a black
      cat.
Parsed sentence:
  (ROOT
    (S
    (NP (DT The) (JJ brown) (NN
       dog))
    (VP (VBZ chases)
      (NP (DT a) (JJ black) (NN
         cat.)))))

Typed dependencies:
  det(dog-3, The-1)
  amod(dog-3, brown-2)
  nsubj(chases-4, dog-3)
  det(cat.-7, a-5)
  amod(cat.-7, black-6)
  dobj(chases-4, cat.-7)
```

```
Example B (passive voice
    sentence):
Input sentence:
  A black cat is chased by the
      brown dog.
Parsed sentence:
(ROOT
  (S
  (NP (DT A) (JJ black) (NN
     cat))
  (VP (VBD was)
    (VP (VBN chased)
    (PP (IN by)
      (NP (DT the) (JJ brown)
         (NN dog.)))))))
Typed dependencies:
  det(cat-3, A-1)
  amod(cat-3, black-2)
  nsubjpass(chased-5, cat-3)
  auxpass(chased-5, was-4)
  det(dog.-9, the-7)
  amod(dog.-9, brown-8)
  agent(chased-5, dog.-9)
```

Table 6: Examples of parsing sentences with Stanford Parser. Both example sentences tell the same story, but in different sentence voice.

The parsed output indicates which words in a sentence belong to same structural groups (e.g. noun or verb phrases). Both the words in the group and the groups themselves are given POS tags. For example, in Example A, the word “dog” has been given a POS tag `NN`, which in Penn Treebank refers to “noun, singular or mass” (Marcus et al., 1993), and the words “the”, “brown”, and “dog” are detected to belong to the same group `NP`, which is a tag for a noun phrase. The typed dependence analysis, on the other hand, shows the relations between words. In Example A, words “chases” and “dog” form a relation `nsubj`, and “chases” and “cat” a relation `dobj`. The relation `nsubj` refers to *nominal subject* (De Marneffe and Manning, 2008a), the syntactic subject of a clause; in other words, it tells, that “dog” is the subject of this sentence. The relation `dobj` refers to *direct object* (De Marneffe and Manning, 2008a). That is, the “cat” is the object of this sentence’s action “chases”. The Example B shows the outcome of an passive voice sentence. Some of the evaluation metrics (e.g. in Introduction) need to detect sentence voices. It can be done by diagnosing the typed dependencies of words in the sentence. The relations `auxpass`, `csubjpass`, and `nsubjpass` indicate that the sentence or clause has passive voice. A full list of relation tags and definitions that the

parser uses, is given in De Marneffe and Manning (2008a).

5 A Study on the User Experience of Scientific Writing Assistant

As a part of this thesis, and for assessing the user experience of Scientific Writing Assistant, we conducted a survey research. Our questionnaire consisted of 31 questions. The first six concerned basic demographic information about the participant (English language nativeness, occupation, and academic background). A total of seven questions were used to clarify participant's background in scientific research: his/her professional research experience, publication history and in general, how much time he/she spents in scientific writing activities. The rest of the questions (18 pcs.) concerned participant's experience in using SWAN, e.g. how much he/she agreed with the results they got from SWAN, or how hard it was to use SWAN. Appendix 4 contains the questions and their answer alternatives.

The aim for this study was to find out

1. How useful SWAN is for the scientific writers?
2. How the users of SWAN experience the tool?

We were also interested in finding out, if there were any differences in experienced usefulness and usage between more the experienced participants versus novices, and between those who had attended Lebrun's writing course versus those who had not.

The survey form was created with Google Forms[9]. The lecturer of *Scientific Writing Skills* -course shared the link to the participants at the last day of the course, and asked participants to fill the survey. Filling it was voluntary. Afterwards, we also added the link to Scientific Writing Assistant, allowing any SWAN user to fill the survey.

After a suitable time, we collected the answers from the questionnaire. We imported the answers of the question form to a spreadsheet program, in which we did basic data cleaning and analysis. We also did some regrouping: to even the professional research experience groups for comparisons, we merged the original "4-6 years" (N = 9), "7-10 years" (N = 6), and "Over 10 years" (N = 5) groups together. As the main questions of our data used Likert scale for their answers, we decided that the percentage

[9] `https://docs.google.com/`

distributions, median, mode, lower and upper quartile were the most suitable methods for summarizing the answers.

5.1 Results

During the time period of August 5th 2012 and April 25nd 2013, participants from at least four different course groups (one in Joensuu, Finland; and three in Singapore) had filled our questionnaire. Some participants answered they had participated in class, but did not specify where (and when). In addition to this, we received answers from SWAN users who had not participated in any of the courses. The total amount of answers was 65.

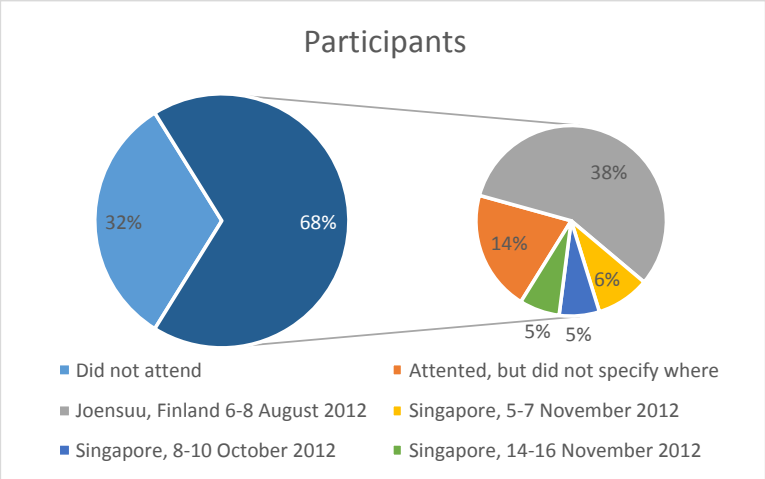

Figure 11: Answers from participants

Participants' demographic data

The majority of our participants are non-native English speakers: 75 % of them are non-natives, and 25 % natives. They are academically highly educated: 57 % has Master's degree, and 23 % has Doctor's degree. They come from various fields of science: computer science, chemistry, physics, medicine, English language, education, and psychology, to name a few. The majority of participants (approx. 70 %) have up to three years of professional research experience (0-1 years: 35 %; 1-3 years: 34 %), although 31 % have 4 to over 10 years of experience. Figure 12 shows these in more detail. Also, the majority (71 %) tells that they have participated in publishing a paper either in local or international journals or conferences.

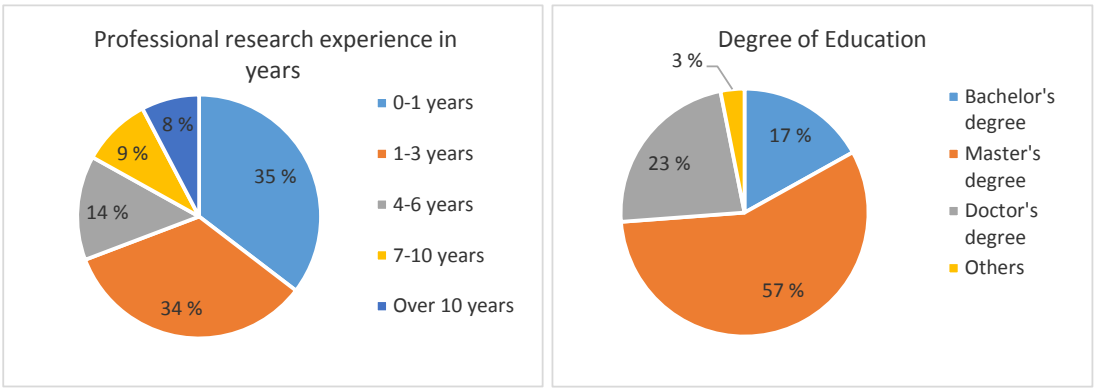

Figure 12: Basic data about participant

Participants' scientific writing activities

We asked the participants about how their time is spent in different writing activities. While a third (31 %) reported they dedicate only small amount (up to 10 %) of their working time in scientific writing activities, 38 % dedivates considerate (10 to 40 %), 18 % large (40 to 60 %), and 12 % very large (60 to 100 %). Of their time dedicated to writing, over half (51 %) of the participants spents moderate time in improving the readability of their text, while for a 22 % the amount of time is remarkable.

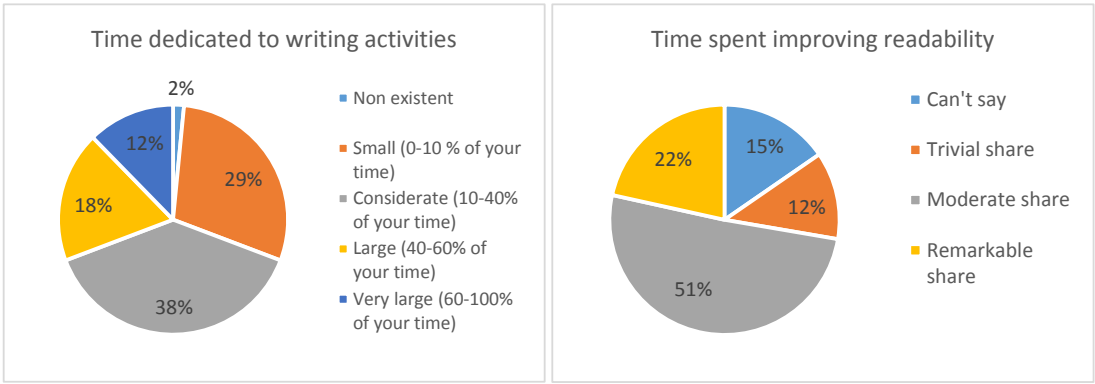

Figure 13: Participants' scientific activities

We also asked participants to identify in which parts of scientific writing they had difficulties. Two of the major problems were "presenting the purpose and goal clearly and interestingly" (25 % of all answers), and "writing fluidly" (21 %). The least problematic parts were "making illustrations" (7 %), and "reporting the experiments" (6 %).

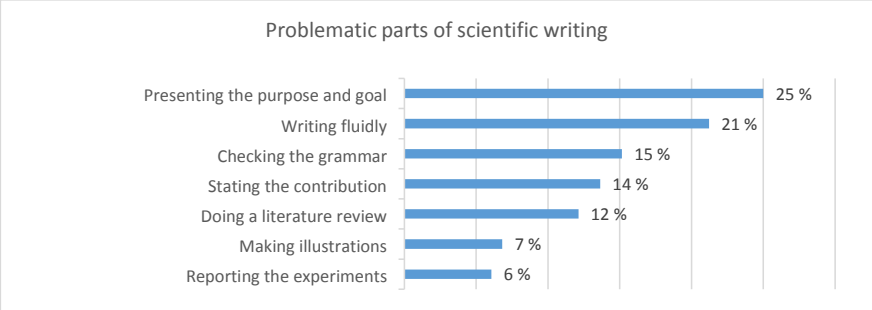

Figure 14: Problematic parts in scientific writing

Participants' agreement with SWAN

We used a Likert scale from 1 (I do not agree at all) to 5 (I agree completely) to investigate participants' agreement with the evaluation results they received from using SWAN. As a whole, both the median and mode values for agreement were 4. No one disagreed completely with the results they received, and two participants even agreed completely with their results (from experience groups 7-10 years, and Over 10 years). By regrouping the experience groups (0-1 years, 1-3 years, 4-6 years, 7-10 years, and Over 10 years) to similar sized units, 0-1 years (23 participants), 1-3 years (22 participants), and 4-10+ years (20 participants), the results for agreement are the following: for all groups, median value was the same as to all groups together, 4; the mode value was for all groups also 4 (for 4-10+ years, alternatives 3 and 4 received equal amount of votes). Figure 15 presents these results.

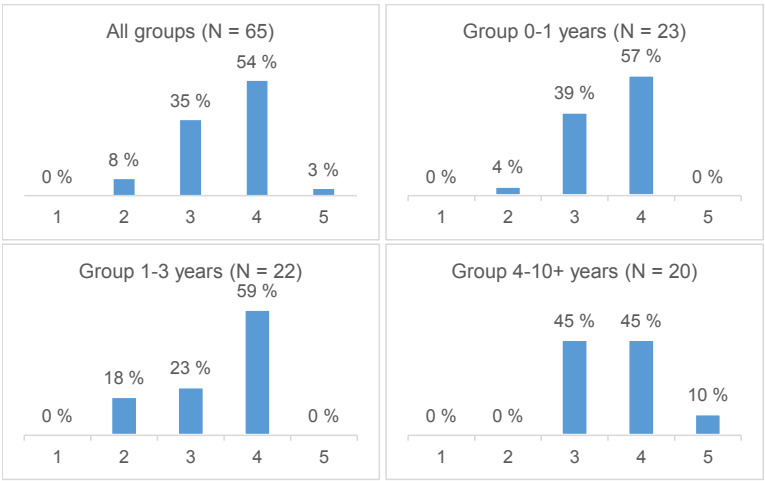

Figure 15: Participants' agreement with SWAN by professional research experience in years.

Participants' usability experience with SWAN

We asked the participants how they experienced SWAN's usability. These questions included asking how hard it was to use SWAN, where they had problems, and whether they understood how their results were calculated.

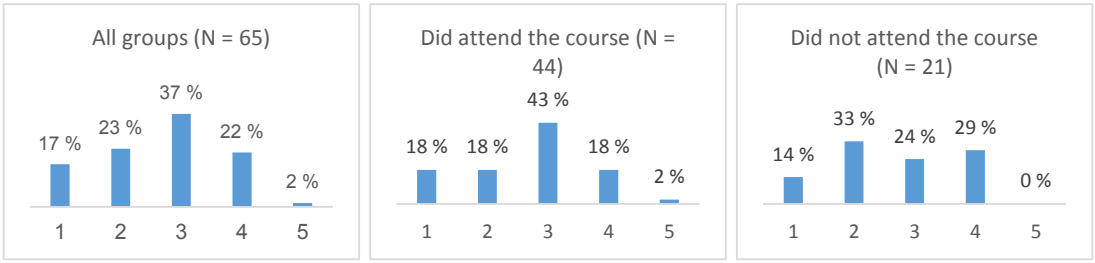

Figure 16: Participants' difficulties with SWAN measured with scale from 1 (Very easy) to 5 (Very difficult)

By using a Likert scale from 1 to 5, we asked how difficult to use SWAN was (Figure 16). The first option (1), in this case, meant that SWAN was very easy to use, whereas the last option (5) meant SWAN was very difficult to use. When viewed by all participants, the most frequent (37 % of the participants) answer was three (3), which as the scale's center item corresponds to neither hard nor easy. The items next the center item received nearly identical amount of votes (2: 23 % vs 4: 22 %). On the other hand, the extreme answers were not as even: while 17 % experienced SWAN very difficult to use, only 2 % thought it was very easy to use. Figure 16 and Table 7 also show participants' answers depending on whether they had attended to Lebrun's course before filling the survey.

Table 7: Participants' difficulties with SWAN. The table shows statistics for different groups.

Group	**Median**	**Mode**	**Quartile 25 % (Q_1)**	**Quartile 75 % (Q_3)**	**IQR**
All	3	3	2	3	1
Did attend the course	3	3	2	3	1
Did not attend	3	2	2	4	2

To assess what kind of problems participants' had with SWAN, we asked whether they understood how to use SWAN, and to identify problems they encountered by selecting the appropriate options from a prepared list, and if not among these, to input their own.

There were no restrictions as to how many, or what selections a participant could make. The options included, among other things, the following: no major problems, problems with importing, problems with starting the program, and problems in understanding the evaluation results. The most frequent answers were: no significant problems (25 %), problems with importing paper and/or structure (17 %), and problems with starting the program (14 %). Three participants reported they had other problems that were not in the prepared list; these concerned loading a previously saved session, problems due to unintuitive user interface, and problems with Java.

Table 8 shows the problematic parts, in addition to all participants, by those who had attended and by those who had not attended to Lebrun's course prior to filling the survey. Those who had not attended to the course, and thus had less prior knowledge about the principles behind SWAN and no immediate help available, had more problems with using the program, compared to those who had attended: importing paper and/or structure (23 % vs 14 %), modifying information to paper (16 % vs 9 %), and with evaluation (6 % vs 3 %). They also considered using SWAN less intuitive (19 % vs 28 %).

Table 8: Problems with SWAN

Problems	Percentage of answers to each problem by		
	All participants	**Attended the course**	**Did not attend the course**
No significant problems (was intuitive)	***25 %***	***28 %***	19 %
With importing paper and/or structure	17 %	14 %	***23 %***
With starting the program	14 %	14 %	13 %
In understanding the evaluation results	13 %	14 %	10 %
With how to begin using the program	13 %	13 %	13 %
With modifying information to paper	12 %	9 %	16 %
With evaluation	4 %	3 %	6 %
Other / With loading previous session	1 %	2 %	0 %
Other / Problems due to unintuitive UI	1 %	2 %	0 %
Other / With Java	1 %	2 %	0 %

Participants' experienced usefulness of various parts of SWAN

To assess the performance and usefulness of SWAN as a tool for improving the quality of scientific manuscripts, we posed a set of questions. We asked both general questions

about the performance of SWAN and which part was most and least useful, and more detailed questions with which we wanted to find out which features in each individual part were most and least useful.

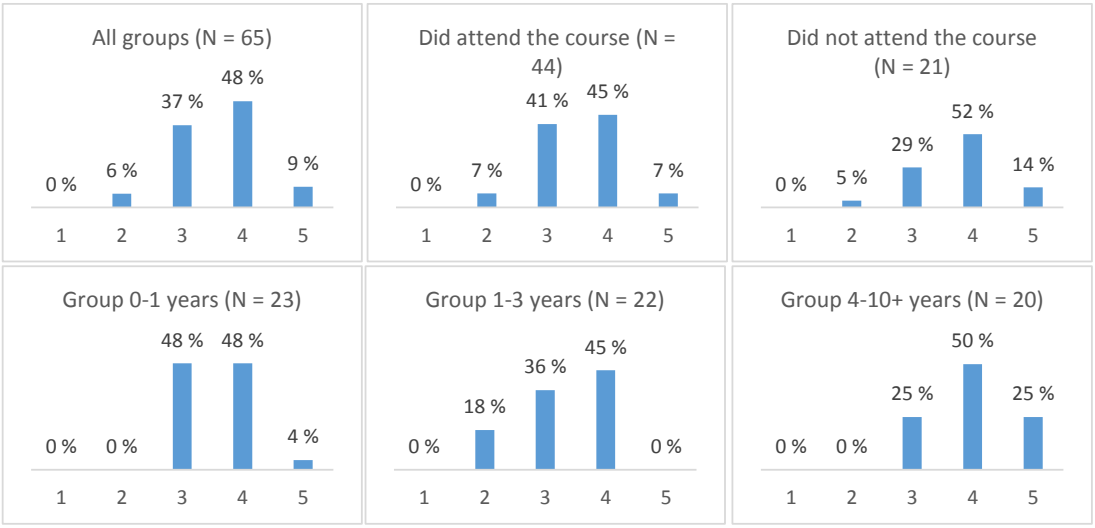

Figure 17: General performance grouped by different units of participants. The answer scale was from 1 (Poorly) to 5 (Very well).

With a question that uses Likert scale from 1 (Poorly) to 5 (Very well), we asked how well in general SWAN performed in its aim to improve writing's quality. The results for this are presented in Figure 17, and in Table 9. For all participants, the most frequent answer was 4. This also applied when participants were regrouped either by attendance to course, or by their professional research experience, excluding the "0-1 years of experience" group, which answered to 3 and 4 equally. Of the experience groups, the one with the most experience (from 4 to over 10 years), voted more on the higher end of the scale (Q_1: 3,75 vs 3,00; Q_3: 4,25 vs 4,00), than the other experience groups.

Table 9: Participants' answers for the general performance of SWAN. The table shows statistics for different groups.

Group	**Median**	**Mode**	**Quartile 25 % (Q_1)**	**Quartile 75 % (Q_3)**	**IQR**
All	4	4	3	4	1
By attendance					
Attended the course	4	4	3	4	1
Did not attend	4	4	3	4	1
By experience					

0-1 years	4	4	3	4	1
1-3 years	3	4	3	4	1
4-10+ years	4	4	3,75	4,25	0,5

In order to get a picture of which parts are considered the most and least useful in SWAN, we posed questions, in which the participant could multi-select the parts they felt the most and least useful. Figure 18 presents, which part were considered most useful, while Figure 19 shows, which parts participants considered the least useful. Table 10 shows the percentages and actual amount of votes each part received in both "most useful", and "least useful" questions; it also shows difference between those votes.

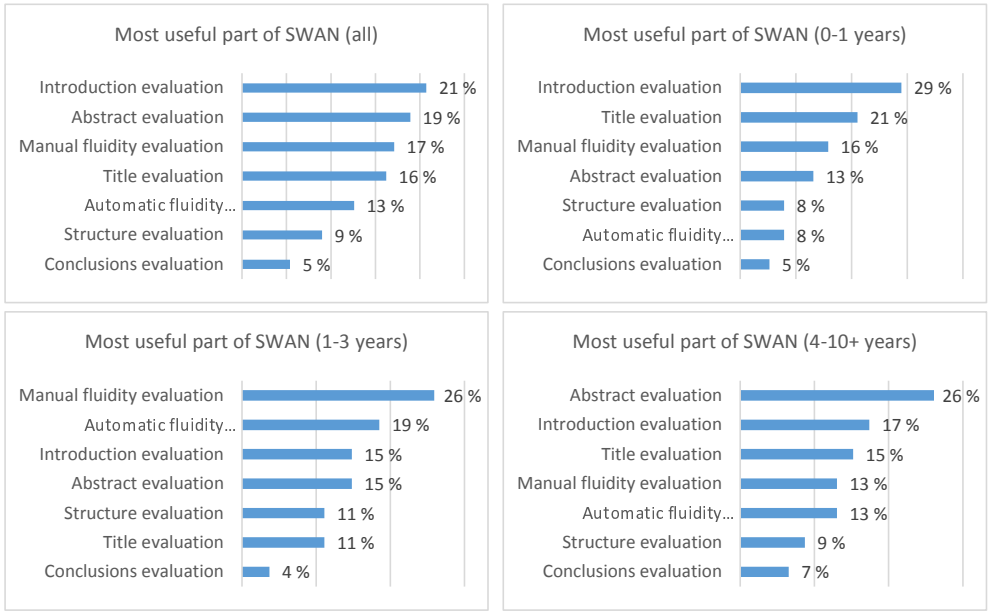

Figure 18: Most useful features

Table 10: Usefulness of the parts in SWAN

Part	Most useful (% and actual values)	Least useful (% and actual values)	Difference between votes
Introduction evaluation	21 % (23)	1 % (1)	22
Abstract evaluation	19 % (21)	6 % (4)	17
Title evaluation	16 % (18)	9 % (6)	12
Automatic fluidity evaluation	13 % (14)	13 % (9)	5
Manual fluidity evaluation	17 % (19)	25 % (17)	2
Conclusions evaluation	5 % (6)	9 % (6)	0
Structure evaluation	9 % (10)	21 % (14)	-4

Other / Visual evaluation	-	1 % (1)	-
Other / Can’t say	-	1 % (1)	-
Other / Everything was useful	-	12 % (8)	-

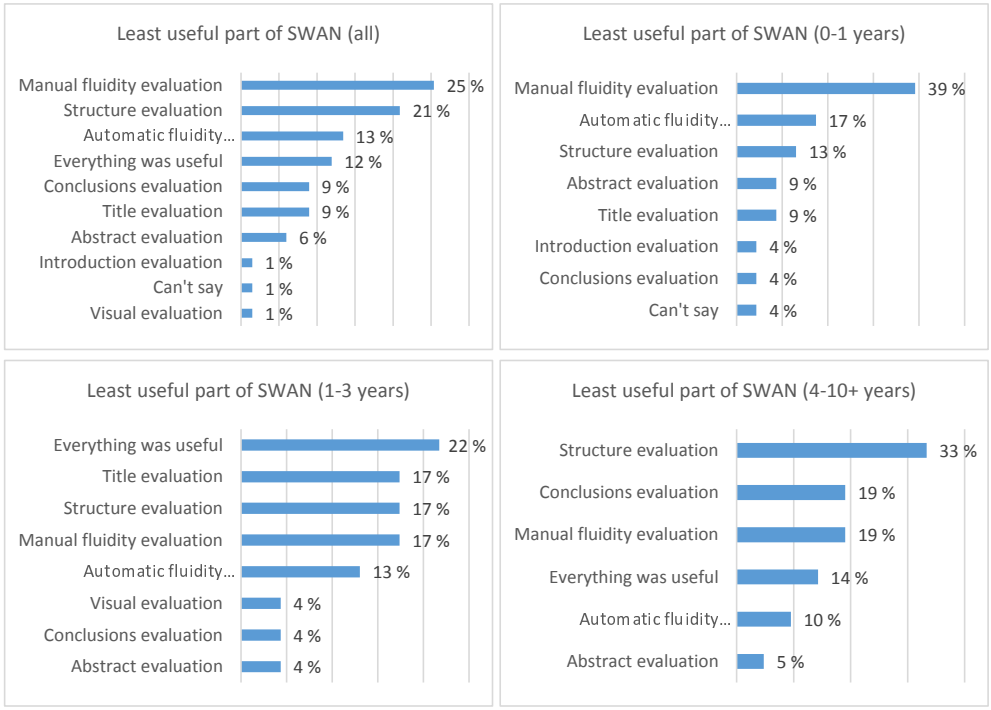

Figure 19: Least useful features

5.2 Discussion

Participants’ agreement with SWAN

Overall, participants agreed with the evaluation results they received from SWAN. They considered the feedback they got reasonable, and helpful: only 8 % voted to disagree, and none disagreed completely. Over a third (35 %) neither agreed nor disagreed, the majority (54 %) agreed, and few (3 %) agreed completely. The agreement was consistent between different professional research experience groups: both inexperienced, and experienced participants agreed with their feedback (mode and median values were 4, and lower, Q_1, and upper, Q_3, quartiles 3 and 4 for all). The participants from the most experienced group, 4-10+ years, did not disagree with their results at all, and 10 % of them even agreed completely. However, unlike the other groups, the majority of them did not agree with their results, instead they divided their votes

between “neither agreed nor disagreed” and “agreed” option. Thus, they seemed to be moderately positive with their results.

Participants’ usability experience with SWAN

In addition to finding out how much participants agreed with the feedback they receive from SWAN, we wanted to see how difficult SWAN is to use. As a whole, the participants found SWAN neither easy nor hard to use. With the Q_1 being 2, and Q_3 being 3, the scale leans to the “easy to use” side.

To see whether there were differences between those, who had attended to Lebrun’s course, and those, who had used SWAN without prior information, we divided the answers by attendance. Somewhat surprisingly, the participants that had attended, seemed to found, as a group, SWAN a bit harder to use (Figure 16): their mode value was 3, while for those who had not attended, it was 2 (the scale was from 1: very easy to 5: very difficult). However, when the results are viewed by lower and upper quartiles (Q_1 and Q_3), they indicate a different story: the Q_3 being 4 for those who had not attended versus 3, for those who had, indicates that the former group answered more to the “difficult to use” side of scale.

We also asked the participants to identify with which parts they encountered problems with. As a whole, the most frequent answer was, that they did not encounter any significant problems, and that SWAN was intuitive to use (25 % of answers). The most problematic parts concerned importing the paper and its structure into the program (17 %), and starting the program (14 %). They also did not always understand the evaluation results (13 %), how to begin after the program was started (13 %), or how to modify information to their paper in SWAN (12 %). Those, who had not attended the course, encountered more problems (Table 8): option “no significant problems (was intuitive)” received 19 % of their answers versus 28 % from those had attended. They also encountered more problems with importing the paper and its structure (23 % vs 14 %), and with modifying information to the paper (16 % vs 9 %).

The results suggest that users are somewhat confused of how to use the program. Importing the paper and its structure requires manual work, and although the program tries to help the user, it seems that the feature is not clear enough for them. SWAN, as a Java program, does not integrate to the operating system it is run as well as some

native programs (e.g. it does not contain an installer, nor does it create shortcuts to the desktop), the users seem to have some difficulties in starting the program. Some of this may be due to Java technology: we have received reports that the users have installed unsuitable versions of Java for their personal computers, which have cause problems with SWAN. Some users have also launched SWAN from the wrong file. To decrease confusion, we have made some modifications as to which files are immediately visible to the user, and from which files the program can be launched. We should also consider, whether we can integrate SWAN better to the operating system it is run.

Participants' experienced usefulness of various parts of SWAN

The participants evaluated the general performance of SWAN as a tool for improving the quality of scientific manuscripts, to be well (Figure 17). We divided the answers by attendance, and by professional research experience. Both those who had attended to Lebrun's course, and those who had not, considered SWAN to perform well. The same applies to the experience groups: the most frequent answer for all, but the "0-1 years" group, was 4 ("well"). The most experienced group (4-10+ years) also considered SWAN to perform better, than the other experience groups. From these answers, it seems, that SWAN is universally among the participants regarded a well performing tool.

The questions, in which the participants were asked to vote for the most and least useful parts of SWAN, reveal that the metrics and results from Introduction, Abstract, and Title evaluation are considered the most useful (Table 10). Those parts received much positive votes (most useful), and less negative votes (least useful). The automatic fluidity was, as the fourth useful part, at the center of the ranking: although it is the most sophisticated metric in the program (see Section 3.6), it was not considered as useful as one might think (writing fluidly was the second most problematic part in scientific writing; see Figure 14). One reason for this might be, that understanding the results and the principle behind the metrics requires some background information. The three least useful parts were (from the least useful): Structure evaluation (9 % of the most useful, and 21 % of the least useful votes), Conclusions evaluation (5 % of most useful, and 9 % of least useful votes), and Manual fluidity evaluation (17 % of most useful, and 25 % of least useful votes). The structure evaluation might be associated with the structure import process, which is considered as the most problematic part of SWAN,

and requires a lot of manual work. The manual fluidity uses the same basic principles as its automatic equivalent, but additionally requires manual effort, which may be the part of the reason of why it is ranked low.

Of the study

Overall we consider the study to be successful. This was the first time we actively asked SWAN users to tell their opinions about SWAN: although we, prior to this study, already had a questionnaire form included in the SWAN, we did not actively ask users to fill it, and did not study the answers we received. This older questionnaire form was used as a basis for the one used in this study. By analysing the data, we found out pointers to further development.

There were, however, few things we could have improved:

- Due to limited time, we did not have much time to prepare the questionnaire form. The questionnaire form was also developed prior to exploring the literature. Thus, some of the questions (those that regarded the usefulness of different parts of SWAN) and their answer alternatives could have been more accurate. We could have also added further questions about the problematic parts of SWAN to get a more accurate view.
- Also, due to the limited time and practical issues, we could not pilot test our questions. With a pilot test we might have been able to test the validity of questions, and identify those that needed revising or that should have been left out. For instance, we had two questions with which we wanted to find out if a participant had participated in Scientific Writing Skills -course. The first of them asked attendance, and the second where and when the course took place. The latter question was optional. The result was that some (N = 9) participants answered they had attended the course, but did not specify where and when. Thus, we were unable to compare answers between different course groups as we considered the other too small ($N_{Singapore}$ = 10; $N_{Joensuu}$ = 25).
- Due to accident, we did not change, right after putting the new questionnaire online, the questionnaire form link in SWAN from the old to the new one. Therefore, some of the participants from the first few courses answered to the old questionnaire instead of the new one. The questionnaires differ somewhat from

each other and their answers are not entirely comparable; thus we were unable to use answers from 46 participants.

6 Conclusions and future work

In this thesis, we introduced Scientific Writing Assistant (SWAN), a computer-based tool that can be used to assess, and improve the quality of scientific manuscripts. SWAN was designed by Lebrun (2011), and developed at the School of Computing, University of Eastern Finland. The project started on 2009, and to this date (2013), continues actively.

The current implementation for SWAN was developed using Java programming language. Thus, SWAN is a multiplatform application, and can be run in any operating system, that supports Java runtime environment version 1.6 (or newer), and graphical desktop system.

We performed a study on the users of SWAN in order to find out how useful and usable SWAN is experienced. Our findings from the study indicate that SWAN users generally agree, and find the feedback they receive from the tool, useful. Further, our study revealed that, while it was generally deemed easy to use, SWAN remains to have some usability problems: users had difficulties in running the tool, beginning to use it, and importing their papers. Generally the metrics, and feedback from the Introduction section were considered the most useful features in the tool. Also, the Abstract and Title metrics were near to the Introduction in the usefulness ranking. The structure metrics were, perhaps due to the laborious nature of the structure import process, considered as the least useful feature.

There are not many computerized tools focusing on purely assessing and/or improving the quality of scientific texts. Instead, most of the tools have focused on a particular audience, for instance on students and their essays. Due to this, and according to our findings from the study we performed, we believe, that SWAN has potential to become a valuable tool for any, who is engaged in scientific writing.

Future work

Next, we will list some suggestions on the short-term, and longer-term work to further improve Scientific Writing Assistant.

Improve the bug diagnostics. SWAN logs, and gives users a chance to send an error

stacktrace on crash. However, the diagnostical message that is automatically generated, does not necessarily contain any specific information of the error, thus complicating debugging. We suggest, that the logging system to be improved to contain more detailed information of a) on which metric, and b) on which particular place at the code the execution is on when the crash occurs. Also, the log system could log other metric-related diagnostical information about the inputted text, e.g. if the metric uses word counts, the logger could include information of the word count.

Integrate WordNet to SWAN. Currently, SWAN is unable to get the root form from a word. This causes problems with matching words with different word classes. In addition to getting the root forms, *WordNet* (Miller et al., 1990) is able to return both the *hypernyms* (a more general class of word; e.g. a hypernym for "a dog" could be "a mammal"), and *hyponyms* (a more detailed class word; e.g. hyponym for "a mammal" could be "a dog"). This would enable SWAN to do semantic-based detection and comparison. The fluidity metrics would benefit most from this integration as they heavily depend on finding common words between sentences. The fluidity algorithm, however, sometimes fails in this, as is described in Example 7. In this example, if the algorithm would have been able to return the hypernyms for the "lion" and the "cat", it could have matched the hypernym "feline" for the words. An example of the results for processing these words with WordNet are described in Table 11.

```
S: (n) lion, king of beasts,
    Panthera leo
  direct hypernym / inherited
     hypernym / sister term
  S: (n) big cat, cat
    S: (n) feline, felid
      ...
      S: (n) mammal, mammalian
        ...
        S: (n) animal, animate
           being, beast, brute,
           creature, fauna
          ...
          S: (n) entity
```

```
S: (n) cat, true cat

  direct hypernym / inherited
     hypernym / sister term
  S: (n) feline, felid
      ...
      S: (n) mammal, mammalian
        ...
        S: (n) animal, animate
           being, beast, brute,
           creature, fauna
          ...
          S: (n) entity
```

Table 11: Results for returning the hypernyms for the words "lion" (on the left) and "cat" (on the right) with WordNet[10]. Both words share common hypernyms, with which they could be matched in word comparison. For simplicity, some less essential tree-levels has been omitted for this example.

[10]WordNet online version (accessed 26.05.2013): `http://wordnetweb.princeton.edu/perl/webwn`

Upgrade the Stanford NLP libraries. The versions of Stanford POS Tagger and Parser, SWAN currently uses are from 2008; the release history from Stanford NLP page[11] lists speed, and accuracy improvements, as well as thread safeness on their later releases. The thread safeness would be especially useful in our software: some NLP processing, such as identifying passive sentences takes long time; with multiple threads we could accomplish speed improvements.

Improve the usability of SWAN. The study we performed indicated several usability problems. The following list contains suggestions based on the results.

- To make starting SWAN less confusing, we suggest providing an installer type of distribution mechanism, instead of the current zip-distribution. This could give us at least two benefits: 1) The users of MS Windows and Apple OSX are accustomed to having their software provided with an installer; thus SWAN would integrate better to the operating system, 2) the installer could, depending on the operating system, hide unnecessary files, and create a shortcut to the desktop; thus users would have a clear single point, from which they can start the tool.
- Improve the help documentation to support users more. The help pages could, for example, include a tutorial with an example paper, and a "frequently asked questions" (FAQ) section.
- According to our study, the document and structure import are considered difficult. However, our study did not specify what makes the import process difficult; thus we suggest studying this issue in more detail (see the next suggestion), and then applying the findings.
- To identify more usability problems, we suggest conducting an usability research. The study, that was described in this thesis focused more on the general, and metric-side aspects of SWAN, and allowed only minor focus on the usability. To examine the actual usability, we would have to design a new survey, that has its main focus on usability, and that would recognize the different attributes of usability: easy to learn, efficient to use, easy to remember, few errors, and subjectively pleasing (Nielsen and Hackos, 1993). In addition to questionnaire and interview type research, the usability research could include inspection methods, such as *heuristic evaluation*, *cognitive walkthroughs*, and *feature inspections* (Nielsen, 1994).

[11] `http://www-nlp.stanford.edu/software`

Refactor the program code. We have not conducted any formal code auditing to SWAN (nor was it in the scope of this thesis); yet, according to the informal communication between developers and our personal experiences, the code quality leaves room for improvement. A poorly written code slows down development (as a poorly written scientific text slows down reading), and increases the risk of defects (Martin, 2008). We suggest familiarizing to principles of clean code as introduced e.g. in Martin (2008), and applying these principles to refactoring. Feathers (2002) also lists things that should be considered when refactoring. One of these is generating test cases prior to refactoring.

Introduce testing more closely to development. So far, the testing during development has been informal, and whether it has been given enough attention, is questionable. We do not suggest any rigid testing procedure, as it would require too much time from the volunteer-based and part-time developing team; however, even a lightweight testing, as long as it is regular and consistent, may decrease the risk of introducing bugs. Generating test cases prior to any larger refactoring, or library update is also recommended.

Suggestions for improvements in various metrics. The following list contains suggestions to the current metrics.

Fluidity The fluidity metric performs only sentence-level fluidity checking. As described in Section 3.6, the fluidity is affected also by the structure of the sentence, for instance by the subject-verb separation (the more words there are between the subject and its verb, the less readable the sentence may be). In addition to this, the progression between the paragraphs also have impact on the fluidity. We suggest to consider including these two factors to the fluidity metrics.

Introduction Questions in the Introduction can increase its attractiveness (see Section 3.3). Therefore, the metric described in Listing 20 searches occurrences for questions. The metric, however, only considers direct questions, such as “What would be, given these requirements, the best way to achieve the aim?”, and ignores implicit questions such as “Given these requirements, we wondered the best way to achieve the aim.”. To make the metric more accurate, we suggest modifying the metric to consider, to some extent, also the implicit questions.

Conclusions As described in Section 3.4, conclusions should contain mention of 1) impact and results of a research, 2) scope and limitations in which research hy-

pothesis works, and 3) potential future work. Currently, the metrics only focus on the future work part, and use word count to determine the conclusions completeness. According to our study, the Conclusions metrics were ranked as second to least useful feature. One way to improve the usefulness for Conclusions metrics might be to include also the first two parts to the metrics. The user could, for instance, be asked, similar to the Title, Abstract and Introduction metrics, to identify the sentences containing the three parts mentioned above.

Structure and Conclusions Make the structure more flexible. The paper structure between the fields of science, and between journals, vary. For instance, some journals and subjects do not use distinct Conclusions section, but rather have the conclusions integrated to the Discussion (see Section 3.4). The feedback we have received indicates that the audience coming from such fields could benefit more from the metrics if the structure could be made more flexible, and configurable.

References

Alley, M. (1996). *The Craft of Scientific Writing*. Springer.

Aluísio, S. M., Barcelos, I., Sampaio, J., and Oliveira Jr, O. N. (2001). How to learn the many unwritten "rules of the game" of the academic discourse: A hybrid approach based on critiques and cases to support scientific writing. In *Advanced Learning Technologies, 2001. Proceedings. IEEE International Conference on*, pages 257–260. IEEE.

Baker, P. N. (2012). How to write your first paper. *Obstetrics, Gynaecology & Reproductive Medicine*, 22(3):81–82.

Barrass, R. (2002). *Scientists Must Write: A Guide to Better Writing for Scientists, Engineers and Students*. Routledge Study Guides Series. Taylor & Francis Group.

Burstein, J., Chodorow, M., and Leacock, C. (2004). Automated essay evaluation: the criterion online writing service. *AI Magazine*, 25(3):27.

Chuck, J.-A. and Young, L. (2004). A cohort-driven assessment task for scientific report writing. *Journal of Science Education and Technology*, 13(3):367–376.

Collins-Thompson, K. and Callan, J. (2004). A language modeling approach to predicting reading difficulty. In *Proceedings of HLT/NAACL*, volume 4.

Council of Science Editors (2006). *Scientific style and format: the CBE manual for authors, editors, and publishers*. Reston, VA : Council of Science Editors in cooperation with the Rockefeller University Press.

Daneman, M. and Carpenter, P. (1980). Individual differences in working memory and reading. *Journal of verbal learning and verbal behavior*, 19(4):450–466.

Daneman, M. and Merikle, P. M. (1996). Working memory and language comprehension: A meta-analysis. *Psychonomic Bulletin and Review*, 3:422–433.

Davis, M., Davis, K. J., and Dunagan, M. (2013). *Scientific papers and presentations*. Academic Press.

Day, R. (1995). *Scientific English: a guide for scientists and other professionals*. Oryx Press.

Day, R. (1998). *How to write and publish a scientific paper*. How to Write & Publish a Scientific Paper. Oryx Press.

De Marneffe, M.-C. and Manning, C. D. (2008a). Stanford typed dependencies manual. Technical report, Stanford University, http://nlp.stanford.edu/software/dependencies_manual.pdf.

De Marneffe, M.-C. and Manning, C. D. (2008b). The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics.

Eisenhart, M. (2002). The paradox of peer review: admitting too much or allowing too little? *Research in Science Education*, 32(2):241–255.

Elserag, H. (2006). Scientific manuscripts: the fun of writing and submitting. *Gastrointestinal endoscopy*, 64(6):S19–S22.

Feathers, M. (2002). Working effectively with legacy code. *Object Mentor, Inc. Available online at http://www. objectmentor. com*.

Feng, L., Jansche, M., Huenerfauth, M., and Elhadad, N. (2010). A comparison of features for automatic readability assessment. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 276–284. Association for Computational Linguistics.

Gopen, G. and Swan, J. (1990). The science of scientific writing. *American Scientist*, 78(6):550–558.

Gopen, G. D. (2004). *Expectations: Teaching Writing from a Reader's Perspective*. Longman Pub Group.

Kakkonen, T. and Sutinen, E. (2004). Automatic assessment of the content of essays based on course materials. In *Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference on*, pages 126–130. IEEE.

Katz, M. J. (2009). *From Research to Manuscript: A Guide to Scientific Writing*. Springer Verlag.

Kinnunen, T., Leisma, H., Machunik, M., Kakkonen, T., and Lebrun, J.-L. (2012). Swan – scientific writing assistant: a tool for helping scholars to write reader-friendly manuscripts. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 20–24. Association for Computational Linguistics.

Klein, D. and Manning, C. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Klein, D. and Manning, C. D. (2002). Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, 15(2003):3–10.

Körner, A. M. (2008). *Guide to publishing a scientific paper*. Routledge.

Kurmis, A. P. (2003). Contributing to research: the basic elements of a scientific manuscript. *Radiography*, 9:277–282.

Lebrun, J.-L. (2011). *Scientific Writing 2.0 – A Reader and Writer's Guide*. World Scientific Publishing Co. Pte. Ltd., Singapore, second edition.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Martin, R. C. (2008). *Clean code: a handbook of agile software craftsmanship*. Prentice Hall.

McCaskill, M. K. (1998). Grammar, punctuation, and capitalization. *A Handbook for Technical Writers and Editors. NASA SP-7084*, 20.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Minnen, G., Carroll, J., and Pearce, D. (2001). Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223.

Montgomery, S. L. (2003). *The Chicago guide to communicating science*. University of Chicago Press.

Moreira, A., Haahtela, T., et al. (2011). How to write a scientific paper-and win the game scientists play. *Rev Port Pneumol*, 17(3).

Nielsen, J. (1994). Usability inspection methods. In *Conference companion on Human factors in computing systems*, pages 413–414. ACM.

Nielsen, J. and Hackos, J. T. (1993). *Usability engineering*. Academic press Boston.

Ortinau, D. J. (2011). Writing and publishing important scientific articles: A reviewer's perspective. *Journal of Business Research*, 64(2):150–156.

Pitler, E. and Nenkova, A. (2008). Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 186–195. Association for Computational Linguistics.

Pololi, L., Knight, S., and Dunn, K. (2004). Facilitating scholarly writing in academic medicine. *Journal of general internal medicine*, 19(1):64–68.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137.

Rodgers, R. and Rodgers, N. (1999). The sacred spark of academic research. *Journal of Public Administration Research and Theory*, 9(3):473–492.

Rosenfeldt, F., Dowling, J., Pepe, S., and Fullerton, M. (2000). How to write a paper for publication. *Heart, Lung and Circulation*, 9(2):82–87.

Shah, J., Shah, A., and Pietrobon, R. (2009). Scientific writing of novice researchers: what difficulties and encouragements do they encounter? *Academic Medicine*, 84(4):511–516.

Si, L. and Callan, J. (2001). A statistical model for scientific readability. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 574–576. ACM.

Singer, A. and Hollander, J. (2009). How to write a manuscript. *The Journal of emergency medicine*, 36(1):89–93.

Sprague, S., Bhandari, M., Devereaux, P., Swiontkowski, M. F., TornettaIII, P., Cook, D. J., Dirschl, D., Schemitsch, E. H., and Guyatt, G. H. (2003). Barriers to full-text

publication following presentation of abstracts at annual orthopaedic meetings. *The Journal of Bone & Joint Surgery*, 85(1):158–163.

Strunk Jr, W. (1918). *The Elements of Style*.

Szklo, M. (2006). Quality of scientific articles. *Revista de Saúde Pública*, 40(SPE):30–35.

Toutanova, K., Klein, D., Manning, C., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Williams, R. and Dreher, H. (2005). Formative assessment visual feedback in computer graded essays. *Informing Science: International Journal of an Emerging Transdiscipline*, 2:23–32.

Witt, P. A. et al. (1995). Writing for publication: rationale, process and pitfalls. *Journal of Park and Recreation Administration*, 13(1):1–9.

Zeiger, M. (2000). *Essentials of writing biomedical research papers*. McGraw-Hill, second edition.

Appendix 1: Resources used in introduction metrics

This Appendix contains lists of expressions that are judgmental, that overstate, are imprecise or establish transitions. These lists, used in the current implementation of SWAN, are updates to the lists of what were originally mentioned in Lebrun (2011), and are based on the same author's newer experiences from the scientific writing courses he organizes.

Judgmental expressions (see Listing 16)			
cannot	fail	fails	failed
inefficient	incapable of	ignore	is obvious
lack	lacks	lacked	limited
naive	not well	not reliable	not robust
not efficient	not capable of	not able to	not perfect
not smart	not coherent	not detailed	not good
not plausible	overlook	plainly see	suffer
slow	tedious	time-consuming	time consuming
unable to	unreliable		

Overstatements (see Listing 17)			
absolutely	absolute	abundantly	acute
acutely	assuredly	certainly	clear
clearly	completely	conclusive	conclusively
decidedly	definite	definitely	diametrically
doubtlessly	effectively	eminently	emphatically
evidently	exact	exactly	extremely
inconceivable	incredibly	indisputable	indisputably
indubitable	inevitable	inevitably	inherently
interestingly	it is obvious that	must	necessarily
necessary	never	no doubt	obvious
obviously	of course	pure	purely
sure	surely	there can be no question that	total
totally	true	truly	unavoidable
unavoidably	unduly	unequivocally	unmistakably

unquestionably	without doubt

Imprecise expressions (see Listing 18)			
a few	a number of	can	commonly
few	frequent	generally	largely
less	mainly	major	many
may	more	most	often
others	overall	probably	several
some	substantial	the main	the majority of
typically	various	widely	

Imprecise expression / Hedge words (see Listing 18)			
about	almost	apparent	apparently
apparently	appear	appearance	appeared
appears	approximately	arguably	around
assume	assumed	assumes	assumption
barely	basically	believed	can
certain	certainly	certainty	conceivably
consistent with	could	doubtful	estimate
estimated	estimates	estimation	fairly
few	frequent	frequently	generally
hopefully	hopefully	improbable	in general
indicate	indicated	indicates	indication
inferred	infrequent	kind of	largely
likelihood	likely	look like	looked like
looks like	mainly	many	may
maybe	might	more or less	most
mostly	occasional	occasionally	often
overall	partly	perhaps	plausible
possibility	possible	possibly	presumably
presume	presumed	presumes	probability
probable	probably	putative	quite
quite clearly	rare	rather	really
reasonably	seem	seemingly	seems
seldom	should	sometimes	somewhat

sort of	speculate	speculated	speculates
speculation	suggest	suggested	suggestion
suggests	suppose	supposed	supposedly
supposes	tend	tended	tendency
tends	think that	thought that	to be expected
to my knowledge	to our knowledge	to the best of our knowledge	uncertain
unlikely	usually	very	would

Transitional expressions (see Listing 25)			
Additionally	Also,	At the same time,	Alternatively
And,	Besides,	Furthermore,	In addition,
Moreover,	On the other hand,		

Appendix 2: Resources used in structure metrics

This Appendix contains lists of non-informative expressions, standard headings in a scientific paper, and general words and auxiliary verbs. These lists, used in the current implementation of SWAN, are updates to the lists of what were originally mentioned in Lebrun (2011), and are based on the same author's newer experiences from the scientific writing courses he organizes.

Non-informative expressions (see Listing 34)			
a bit	a great deal of	a lack of	acquisition
activity	all	all of	another
any	anybody	anything	application
approach	architecture	bit	both
capability	characterization	comparison	computer
concept	condition	configuration	demonstration
development	discussion	each	each other
effect	either	enough	estimate
estimation	evaluation	everybody	everyone
everything	example	experiment	few
formulation	framework	generation	good
investigation	like	literature review	lots of
many	material	measurement	model
modification	most	most of	much
neither	no	no one	nobody
none	nothing	other	overview
parameter	plenty of	preparation	principle
problem	procedure	process	quantification
related works	results	review	round
save	section	self	set
setup	several	significance	significant
simulation	some	statement	statistical
strategy	study	system	technique
test	that	theoretical	these
this	those	verification	what
which	who		

Standard headings (see Listing 34)			
Abstract	Background	Conclusion	Conclusions
Discussion	Introduction	Materials and Methods	Methodology
Reference	Related works	Result	Results

General words and auxiliary verbs (see Listing 33)			
am	am not	are	are not
be	been	can	can not
could	could not	data	do
do not	is	is not	method
might	might not	must	must not
shall	shall not	should	should not
source	summary	was	was not
were	were not	will	will not
would	would not		

Appendix 3: Resources used in fluidity metrics

Appendix 3A Resource lists

This Appendix contains lists of placebo transitions, and fluid expressions. These lists, used in the current implementation of SWAN, are updates to the lists of what were originally mentioned in Lebrun (2011), and are based on the same author's newer experiences from the scientific writing courses he organizes.

Placebo transitions			
additionally	also	and	another
besides	furthermore	however	in addition
meanwhile	moreover	on the other hand	other
the above mentioned	the former	the latter	the other
to add			

Fluid expressions			
admittedly,	after	afterward	again
all in all	along these lines,	although	as a consequence
as a result,	as expected	as soon as,	be that as it may
because	before	but	consequently
conversely	curiously	despite	during
equally,	even though	eventually	figure
finally	first	firstly	following
for example	for instance	for this reason,	in a certain sense
in a similar	in a way	in comparison,	in conclusion
in contrast	in other words	in particular	in short
in summary	in the first	in the same way	indeed
initially	interestingly	it follows	it is as if,
last	lastly	likewise	meanwhile,
nevertheless	next	nonetheless	now,
once	regardless	similarly	so
so far	specifically,	still,	subsequently,
such	surprisingly,	that is why,	the first

the last	the next,	the reason	then
this	this is why	thus	to conclude,
to elaborate	to explain,	to illustrate	to put it another way
to put it succinctly	to sum up	to summarize,	ultimately
unexpectedly	until	up to now,	whereas
while	while,	yet	

Listing 38 contains requlars expression that are used in finding fluid words at the beginning of sentences. These include alphabet and numbered bullets “a)”, “b)”, “1)”, “2)”, and ordinal numbers “first”, “second”.

```
Alphabet bullets:
[A-Z]+\\): "A)", "B)", "C)", ...
\\([A-Z]+\\): ‘‘(A)’’, ‘‘(B)’’, "(C)", ...

Numbered bullets:
[1-9]+\\): "1)", "2)", "3)", ...
\\([1-9]+\\): "(1)", "(2)", "(3)", ...

Ordinal numbers:
"first", "second", "third", "fourth", ...
[1-9]*1st, [1-9]*2nd, [1-9]*3rd, [4-9]th, [1-9]+0th
```

Listing 38: Regular expressions used in finding fluid words at the beginning of sentences.

Appendix 3B Fluidity algorithm as a pseudocode

The following listing contains pseudocode for the fluidity metric algorithm.

```
1 preProcess(text)
2 firstSentence = paragraph.getSentences().first
3 sentence.setType(NOT_APPLICABLE)
4 setDefaultWordSet(sentence)
5 FOR sentence = paragraph.getSentences().second TO
      paragraph.getSentences().last
```

```
  sentence.setType(UNKNOWN)
  sentence.hasPlaceboTransitions(checkPlaceboTransitions(sentence))
  IF sentenceBeginsWithFluidWords(sentence)
    sentence.setType(FLUID)
    setDefaultWordSet(sentence)
  ELSE
    previousSentence = sentence.getPrevious() //Sn-1 sentence
    checkSentenceProgression(sentence, previousSentence)

    IF sentence.getType() == INVERTED_TOPIC_CANDIDATE
      sentence.setType(INVERTED_TOPIC)
    ELSEIF sentence.getType() == UNKNOWN
      FOR offset = 2 TO 3 //check Sn-2 and Sn-3
        previousSentence = sentence.getPrevious(offset)
           //n-offset sentence
        checkSentenceProgression(sentence, previousSentence)
        IF sentence.getType() != UNKNOWN
          BREAK
    defineResults(sentence)

preProcess(text)
  // use regular expressions to detect references
  removeLiteratureReferencesCitations(text)
  // split by: newline (\n), carriage return (\r) and
     paragraph-separator characters (\u2029)
  paragraphs = splitTextIntoParagraphs(text)
  // use NLP tools & WordSentenceProcessor
  splitAndTagTextInParagraphsIntoSentences(paragraphs)
  removeShortStubs(paragraphs)

defineResults(sentence)
  IF sentence.getType() == UNKNOWN
    sentence.setType(DISCONNECTED)
    setDefaultWordSet(sentence)
  ELSE
    // check from which round topics and stresses should be used
    IF sentence.getType() == INVERTED_TOPIC
      wordSetFromRound = 1 // first round (with Sn-1)
    ELSEIF sentence.getType() IN (FLUID, OUT_OF_SYNC)
      wordSetFromRound = sentence.getOffset() // latest round
    sentence.setStrongTopics(sentence.getStrongTopics(wordSetFromRound))
    sentence.setWeakTopics(sentence.getWeakTopics(wordSetFromRound))
```

```
46      sentence.setStrongStresses(sentence.getStrongStresses(wordSetFromRound))
47      sentence.setWeakStresses(sentence.getWeakStresses(wordSetFromRound))
48
49  removeShortStubs(paragraphs)
50    shortStubStart {TaggedWord("it", "PRP"), TaggedWord("there",
        "EX")}
51    shortStubEnd {TaggedWord("that", "IN")}
52    sentences = paragraphs.getSentences()
53    FOR sentence : sentences
54      IF sentence.startsWith(shortStubStart) AND
          sentence.endsWith(shortStubEnd)
55        startIndex = sentence.indexOf(shortStubStart)
56        endIndex = sentence.indexOf(shortStubEnd)
57        sentence.removeWordsBetween(startIndex, endIndex)
58
59  setDefaultWordSet(sentence)
60    mainClauseSubjects = sentence.getMainClauses().getSubjects()
61    sentence.setStrongTopicsFinal(mainClauseSubjects)
62    nounsAndVerbDerivedNouns = getNouns(sentence) +
        getVerbDerivedNouns(sentence)
63    stressWords = nounsAndVerbDerivedNouns - mainClauseSubjects
64    addStressWords(sentence, stressWords)
65
66  sentenceBeginsWithFluidWords(sentence)
67    beginningWords = sentence.getWordsFromBeginning() // words
        from begin until first verb (excl. gerund form)
68    IF beginningWords.contains(FLUID_WORDS) OR // See Appendix 3A
69      beginningWords.contains(PRONOUNS) OR
70      beginningWords.contains(FLUID_WORDS_REGEX)
71      RETURN TRUE
72    RETURN FALSE
73
74  checkSentenceProgression(sentence, previousSentence)
75    sentence.offset++ // +1 to offset
76    checkSentenceMainClauses(sentence, previousSentence)
77    checkWholeSentence(sentence, previousSentence)
78
79  checkSentenceMainClauses(currentSentence, previousSentence)
80    offset = currentSentence.getOffset()
81    mainClauseSubjects =
        currentSentence.getMainClauses().getSubjects()
```

```
82    prevSentenceTopicsAndStresses =
          previousSentence.getAllTopics() +
          previousSentence.getStrongStresses()
83    matchedWords =
          prevSentenceTopicsAndStresses.getMatches(mainClauseSubjects)
84    IF matchedWords NOT empty
85      topicFound(word, STRONG_TOPIC, currentSentence,
            previousSentence, FALSE, offset)
86    stressWords = subjects - matchedWords
87    addStressWords(currentSentence, stressWords, offset)
88
89  checkWholeSentence(currentSentence, previousSentence)
90    offset = currentSentence.getOffset()
91    reachedVerb = false
92    reachedTopicOrMainSentence = false
93    prevSentenceTopicsAndStresses =
          previousSentence.getAllTopics() +
          previousSentence.getStrongStresses()
94    FOR word : currentSentence
95      reachedVerb = isVerb(word) OR reachedVerb
96      reachedTopicOrMainSentence =
            currentSentence.isMainClauseWord(word) OR
            reachedTopicOrMainSentence
97      IF !reachedVerb
98        matches = prevSentenceTopicsAndStresses.matches(word)
99        IF (matches AND isNoun(word)) OR (matches AND
              reachedTopicOrMainSentence AND isVerbDerivedNoun(word))
100         topicFound(word, WEAK_TOPIC, currentSentence,
                previousSentence, FALSE, offset)
101         reachedTopicOrMainSentence = true
102       ELSEIF isVerbDerivedNoun(word)
103         addStressWords(word, offset)
104     ELSE
105       IF currentSentence.hasTopic(offset)
106         IF isNoun(word)
107           addStressWords(word, offset)
108         ELSEIF isVerbDerivedNoun(word)
109           matches = prevSentenceTopicsAndStresses.matches(word)
110           IF matches
111             addStressWords(word, offset)
112       ELSE
113         matches = prevSentenceTopicsAndStresses.matches(word)
```

```
114         IF matches AND isNoun(word)
115           topicFound(word, WEAK_TOPIC, currentSentence,
                 previousSentence, TRUE, offset)
116           reachedTopicOrMainSentence = true
117         ELSE matches AND isVerbDerivedNoun(word)
118           addStressWords(word, offset)
119
120 topicFound(topicWords, topicType, sentence, previousSentence,
      reachedVerb, offset)
121   IF offset == 1 // checking the Sn-1 sentence
122     IF !reachedVerb
123       sentence.setType(FLUID)
124     ELSE
125       sentence.setType(INVERTED_TOPIC_CANDIDATE)
126   ELSE // checking the Sn-2...3 sentences
127     IF !reachedVerb
128       IF sentencesBetweenFluidOrInvertedTopic(sentence,
             previousSentence)
129         sentence.setType(FLUID)
130       ELSE
131         sentence.setType(OUT_OF_SYNC)
132   IF topicType == WEAK_TOPIC
133     IF !sentence.getStrongTopics(offset).contains(topicWords)
134       sentence.addWeakTopicWords(topicWords, offset)
135   ELSEIF topicType == STRONG_TOPIC
136     sentence.addStrongTopicWords(topicWords, offset)
137
138 addStressWords(sentence, stressWords)
139   FOR stressWord IN stressWords
140     IF isStrongStress(sentence, stressWord)
141       sentence.addStrongStressWords(stressWord)
142     ELSE
143       sentence.addWeakStressWords(stressWord)
144
145 isStringStress(sentence, stressWord)
146   IF isNoun(stressWord)
147     IF appearsBeforeFirstPunctuationMark(stressWord, sentence) OR
148       appearsAfterLastPunctOrConjVerb(stressWord, sentence)
149       RETURN TRUE
150     IF sentence.getMainClauses().contains(stressWord) AND
          sentence.getMainClauses().containsTopic()
151       RETURN TRUE
```

```
152     IF sentence.getPrecedingWord(stressWord).isNumber()
153       RETURN TRUE
154   ELSEIF isVerbDerivedNoun(stressWord) AND
          sentence.getMainClauses().contains(stressWord)
155     RETURN TRUE
156   RETURN FALSE
```

Listing 39: Complete pseudocode for the fluidity metric algorithm

Appendix 4: Questions used in the study

This appendix contains the questions used in the study described in Section 5. Tables 15–19 list the questions, the answer types (single-select, multi-select, Likert scale, or text), the answer alternatives, whether the answer alternatives included “Other” option, into which the participant could freely input text, and whether the question was compulsory.

The total amount of question was 31. Questions 1–6 were about participants’ basic demographic data. Questions 7–13 regarded participants’ scientific writing background. Questions 14–31 considered SWAN.

Table 15: Questions and answer alternatives: 1–9 / 31

		Answer alternatives					
Questions	**Type**	**1**	**2**	**3**	**4**	**"Other" option**	**Required**
1. Are you a native English speaker?	Single	Yes	No			No	Yes
2. What is your degree of education?	Single	Undergraduate	Bachelor's degree	Master's degree	Doctor's degree	Yes	Yes
3. What is the subject of your degree of education (e.g. major subject)?	Text					No	Yes
4. What is your current occupation/job title?	Text					No	Yes
5. What is your background knowledge regarding the book "Scientific Writing: a reader and writer's guide" and Scientific Writing Skills class?	Single	I have not participated to Scientific Writing Skills class by Jean-Luc Lebrun and I am not familiar with his book "Scientific Writing: a reader and writer's guide".	I have studied Lebrun's book but not taken part to his Scientific Writing Skills class.	I have attended to Lebrun's Scientific Writing Skills class.		No	Yes
6. If you ticked the last one of the previous question, you may also indicate where and when you participated to the Scientific Writing Skills course. (Not compulsory)	Text					No	No
7. What percentage of your total working time is dedicated to scientific writing activities?	Single	Non existent	Small (0-10 % of your time)	Considerate (10-40 % of your time)	Very large (60-100 % of your time)	No	Yes
8. How many journal publications have you produced?	Single	0 journal publications	1-3 journal publications	4-10 journal publications		No	Yes
9. How many conference publications have you produced?	Single	0 conference publications	1-3 conference publications	4-10 conference publications		No	Yes

Table 16: Questions and answer alternatives: 10–16 / 31

		Answer alternatives								
Questions	**Type**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**"Other" option**	**Required**
10. Where have you published?	Multi	I have published in international conferences	I have published in local conferences	I have published in international journals	I have published in local journals	I haven't published			No	Yes
11. How many years of professional research experience do you have?	Single	0-1 years	1-3 years	4-6 years	7-10 years	Over 10 years			No	Yes
12. Which parts of the paper writing did you find the most problematic?	Multi	Stating the contribution	Writing fluidly	Reporting the experiments	Doing a literature review	Making illustrations	Checking the grammar (correct tenses, word order, etc.)	Presenting the purpose and goal clearly and interestingly	No	Yes
13. How much time do you use on average to improve the readability of the texts you author compared to overall time it takes to write the texts?	Single	Trivial share	Moderate share	Remarkable share	Can't say				No	Yes
14. How well in general did SWAN perform as a tool for improving the quality of scientific writing?	Likert scale	1: Poorly	2	3	4	5: Very well			No	Yes
15. How difficult was it in general to use the software?	Likert scale	1: Very easy	2	3	4	5: Very difficult			No	Yes
16. Did you understand the path from starting the software to getting evaluation results?	Multi	Yes, it was intuitive	I had problems with starting the program	I had problems with how to begin using the program	I had problems with importing my paper and/or structure	I had problems with with modifying information to my paper	I had problems with evaluation	I had problems in understanding the evaluation results I got	Yes	Yes

Table 17: Questions and answer alternatives: 17–22 / 31

		Answer alternatives								
Questions	**Type**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**"Other" option**	**Required**
17. How much did you agree with the results SWAN gave you?	Likert scale	1: I didn't agree at all	2	3	4	5: I agreed completely			No	Yes
18. Did you understand connections between the metrics and the results you got?	Single	Connections were easy to understand	I understood them most of the time	Connections were hard to understand					No	Yes
19. What was the best part of the software?	Multi	Title evaluation	Abstract evaluation	Introduction evaluation	Conclusions evaluation	Structure evaluation	Automatic fluidity evaluation	Manual fluidity evaluation	Yes	Yes
20. What was the least useful part of the software?	Multi	Title evaluation	Abstract evaluation	Introduction evaluation	Conclusions evaluation	Structure evaluation	Automatic fluidity evaluation	Manual fluidity evaluation	Yes	Yes
21. What kind of information regarding to the title did you find most useful?	Multi	How to use/have title search keywords	How the title relates to other sections of scientific text (e.g. the connection between title and abstract)	How to make the title more clear and attractive	How the contribution should be placed/-considered in your title	Which sections you should have in your title	I did not use this part of SWAN		No	Yes
22. What kind of information regarding to the title did you find least useful?	Multi	How to use/have title search keywords	How the title relates to other sections of scientific text (e.g. the connection between title and abstract)	How to make the title more clear and attractive	How the contribution should be placed/-considered in your title	Which sections you should have in your title	I found everything useful	I did not use this part of SWAN	No	Yes

Table 18: Questions and answer alternatives: 23–26 / 31

		Answer alternatives							
Questions	**Type**	**1**	**2**	**3**	**4**	**5**	**6**	**"Other" option**	**Required**
23. What kind of information regarding to the abstract did you find most useful?	Multi	How the abstract relates to other sections of a scientific text	Which sections should be available in an abstract and in which order they should be	What other elements should be in an abstract (e.g. the use of numbers brings precision to the results in abstract)	How to make your abstract more attractive (e.g. hints about the use of tenses)	I did not use this part of SWAN		No	Yes
24. What kind of information regarding to the abstract did you find least useful?	Multi	How the abstract relates to other sections of a scientific text	Which sections should be available in an abstract and in which order they should be	What other elements should be in an abstract (e.g. the use of numbers brings precision to the results in abstract)	How to make your abstract more attractive (e.g. hints about the use of tenses)	I found everything useful	I did not use this part of SWAN	No	Yes
25. What kind of information regarding to the introduction did you find most useful?	Multi	How to make introduction more personal and engaging (e.g. the use of personal pronouns)	How you should consider the length and variation of sentences/sentence segments	Which words you should avoid in order not to e.g. bring imprecise tone to you introduction	How you should end your introduction in order to make it more interesting	I did not use this part of SWAN		No	Yes
26. What kind of information regarding to the introduction did you find least useful?	Multi	How to make introduction more personal and engaging (e.g. the use of personal pronouns)	How you should consider the length and variation of sentences/sentence segments	Which words you should avoid in order not to e.g. bring imprecise tone to you introduction	How you should end your introduction in order to make it more interesting	I found everything useful	I did not use this part of SWAN	No	Yes

Table 19: Questions and answer alternatives: 27–31 / 31

		Answer alternatives							
Questions	**Type**	**1**	**2**	**3**	**4**	**5**	**6**	**"Other" option**	**Required**
27. What kind of information regarding to the conclusion did you find most useful?	Multi	What should be included in the conclusion text (e.g. few words about future work)	Which tenses should be used and which shouldn't in the conclusion	I did not use this part of SWAN				No	Yes
28. What kind of information regarding to the conclusion did you find least useful?	Multi	What should be included in the conclusion text (e.g. few words about future work)	Which tenses should be used and which shouldn't in the conclusion	I found everything useful	I did not use this part of SWAN			No	Yes
29. What kind of information regarding to the fluidity (automatic) did you find most useful?	Multi	What causes sentence fluidity to be disconnected (e.g. isolated topics)	What the inverted topic is and how it affects the fluidity	What the stress words are and how they affect the fluidity	In general, how to make sentences more fluid	I did not use this part of SWAN		No	Yes
30. What kind of information regarding to the fluidity (automatic) did you find least useful?	Multi	What causes sentence fluidity to be disconnected (e.g. isolated topics)	What the inverted topic is and how it affects the fluidity	What the stress words are and how they affect the fluidity	In general, how to make sentences more fluid	I found everything useful	I did not use this part of SWAN	No	Yes
31. Other feedback	Text							No	No