DISTANCE MEASURES FOR SYNONYMS

DEVELOPMENT PROJECT

SUPERVISOR

Colin Fidge

STUDENT

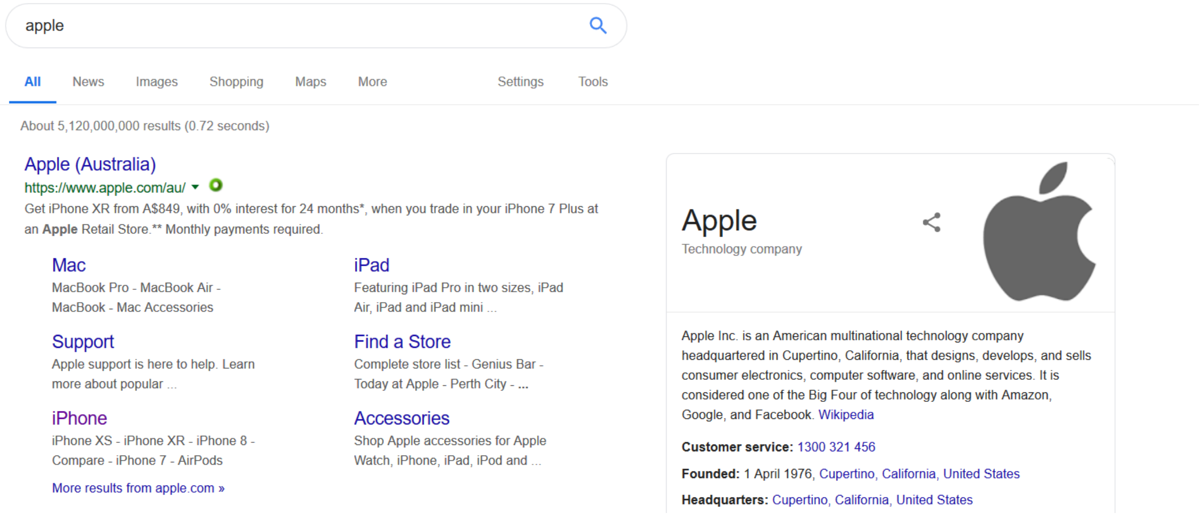
Duy Hai Nguyen

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Background Information

Naturally, in the use of English in everyday lives, there are a number of ways to say the same, similar, or closely-related things. For example, "person" and "human" could be assumed to mean the same concept. However, "residents", "taxpayers" and "citizens", while being closely-related, could be used in different contexts due to the implied social structure underneath their surface meaning. Recently, another interesting linguistic phenomenon has been identified, in which a close semantic relationship can be found between seemingly unrelated concepts. For example, "apple" sometimes refers to the American technology company that produces the world-famous iPhone or iPad.



About 5,120,000,000 results (0.72 seconds)

Apple (Australia)
https://www.apple.com/au/
Get iPhone XR from A$849, with 0% interest for 24 months*, when you trade in your iPhone 7 Plus at an **Apple** Retail Store.** Monthly payments required.

Mac
MacBook Pro - MacBook Air - MacBook - Mac Accessories

iPad
Featuring iPad Pro in two sizes, iPad Air, iPad and iPad mini ...

Support
Apple support is here to help. Learn more about popular ...

Find a Store
Complete store list - Genius Bar - Today at Apple - Perth City - ...

iPhone
iPhone XS - iPhone XR - iPhone 8 - Compare - iPhone 7 - AirPods

Accessories
Shop Apple accessories for Apple Watch, iPhone, iPad, iPod and ...

More results from apple.com »

Apple
Technology company

Apple Inc. is an American multinational technology company headquartered in Cupertino, California, that designs, develops, and sells consumer electronics, computer software, and online services. It is considered one of the Big Four of technology along with Amazon, Google, and Facebook. Wikipedia

Customer service: 1300 321 456

Founded: 1 April 1976, Cupertino, California, United States

Headquarters: Cupertino, California, United States

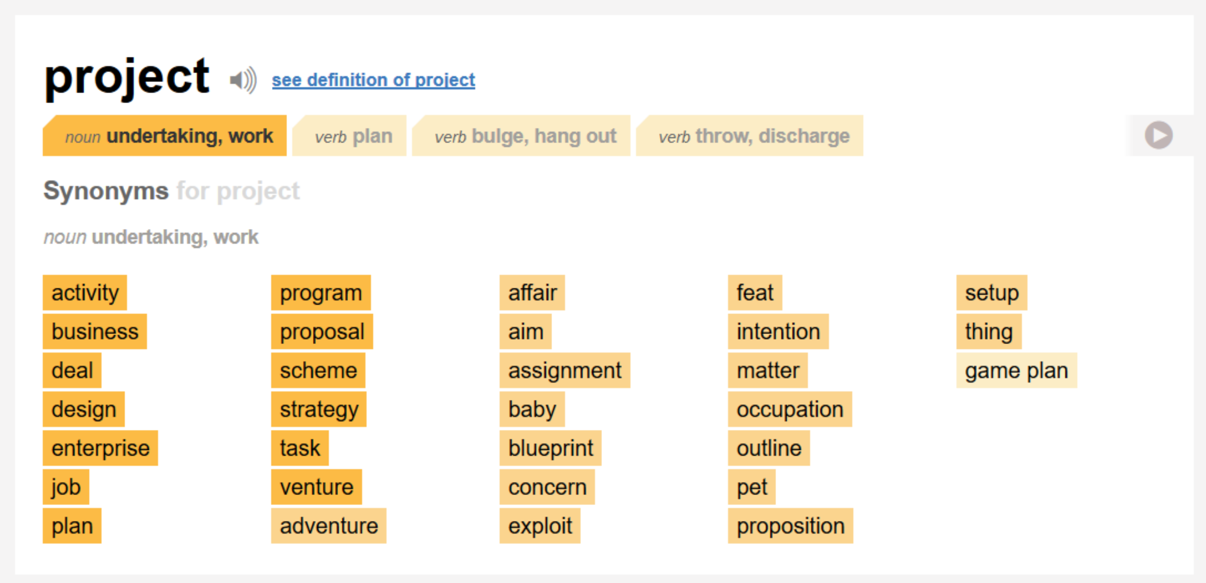
The above examples, to some extent, demonstrate the complication in assessing types of similarity in natural language processing, the two most popular of which are ***syntactic similarity*** and ***semantic similarity***. ***Syntactic similarity*** between two words refers to the visible similarity based on the comparison between the order or presence of characters in each word (Oliva, Serrano, Castillo, & Iglesias, 2011). For instance, "fat" and "fact" have good syntactic similarity as they are only one character away from each other, but they have no semantic similarity as they mean totally different things. On the other hand, ***semantic similarity*** simply describes similarity in meaning between two words (Oliva et al., 2011). For instance, “house” and “dwelling” could be deemed as synonyms due to their clear semantic similarity. Finally, there are cases where a pair of word has both types of similarity, mostly due to the various forms derived from a word. For example, “green” and “greenish” are related both syntactically and semantically.

1.2. Research Problem

After previous literature on the phenomenon above was examined, it is found that a significant number of studies already focus on measuring syntactic string similarity by using various methods such as *n-gram similarity* (Kondrak, 2005), *Levenshtein distance* (Schepens, Dijkstra, & Grootjen, 2012), *Jaccard similarity* (Chaudhuri, Ganti, & Kaushik, 2006), *TF-IDF cosine similarity* (Salton & Buckley, 1988) or *Jaro-Winkler measure* (Wang, Qin, & Wang, 2017). Other improvements built on these measures can include *word stemming*, *subsequence matching*, *part-of-speech tagging*, *stop-word removal* and various other weighting and normalisation approaches (Salton & Buckley, 1988). As these studies mainly investigate syntactic similarity, they can only help measure the visible distance between two words, while being unable to capture any underlying semantic similarities that involve synonyms, antonyms or abbreviations.

Meanwhile, as opposed to the large amount of previous literature on syntactic similarity, research on semantic similarity appears to be insufficient or under-developed. A few studies found in this domain focus on specific topics such as automatic discovery of similar words (Senellart & Blondel, 2004), string similarity measures (Mihalcea & Corley, 2006; Lu et al., 2013), and measuring similarity between words using web search engines (Bollegala, Matsuo, & Ishizuka, 2007).

Although these few studies could be useful in developing a method to measure semantic similarity theoretically, from a practical perspective, there is currently no tangible product that can quantify the similarity between two synonyms or closely-related words. The most popular online tool to loop up synonyms – *Thesaurus* (https://www.thesaurus.com/) – when providing a list of synonyms or related words, only mark the level of closeness of synonyms by colours.



project see definition of project

noun undertaking, work | verb plan | verb bulge, hang out | verb throw, discharge

Synonyms for project

noun undertaking, work

activity
business
deal
design
enterprise
job
plan
program
proposal
scheme
strategy
task
venture
adventure
affair
aim
assignment
baby
blueprint
concern
exploit
feat
intention
matter
occupation
outline
pet
proposition
setup
thing
game plan

Therefore, this project aims to quantify how semantically close two synonyms or related words are by developing a web application which, given a word by the user, will use an online dictionary or thesaurus database to create a ranked list of related words, each of which will include a score to indicate their semantic closeness to the original word.

1.3. Project Objectives

The objectives of the project are threefold. The first one is to research the types of dictionaries, thesauri or other applications that work with synonyms, semantics or related words. The second objective is to investigate the current approaches, measures, methods, algorithms or Application Programming Interfaces (APIs) to extract synonyms or related words from the Internet and establish a framework for the application to be developed. Finally, the project also aims to provide a better understanding of semantic similarity and produce a clear visual demonstration of a variety of relations between words, which can be used as groundwork for future application development in the same field.

1.4. Project Methodology Approach

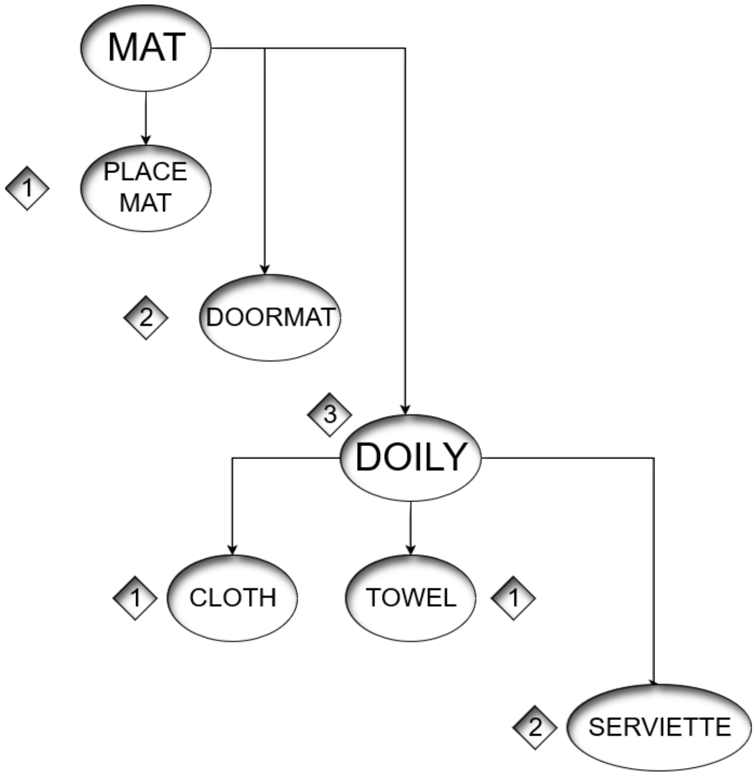
In order to execute the project, previous literature will be examined to assess the current methods, approaches, measures or algorithms in the field of semantic similarity in natural language processing. The acquired knowledge may reveal useful methods in developing a scheme to quantify the closeness between words during the development phase of the project. Additionally, the project involves getting APIs or scraping results from online dictionaries and thesauri. Afterwards, the acquired APIs or scraped data will be used to implement the back-end, design the front-end, and ultimately deliver a user-friendly web application. Although the main outcome of the project is a tangible artefact – a web application in this circumstance, the project will still be research-informed, which means literature or data evidence will be utilised to validate the design and implementation as well as assess the final value or impact of the artefact to be delivered.

1.5. Project Scope

As the project involves working with complicated tasks in natural language processing, its scope is limited to English words only. Also, as mentioned in section 1.2, the project – for now – only investigates semantic similarity as the field is under-developed both theoretically and practically. In other words, syntactic similarity will not be examined during the proposed duration of the project. However, it could be reconsidered to be briefly integrated as an additional feature during the last stages of the project only if the project can still follow its proposed plan and thus spare sufficient time to develop more features related to this type of word similarity.

1.6. Project Deliverables

As the project is a development one, its main deliverable is a web application for users to input a word and retrieve its synonyms or related words, each of which is accompanied by its ranking, score or closeness rating demonstrating its semantic distance to the original word. Furthermore, each of these result words can be subsequently queried in a similar manner to examine the accumulation or consolidation of distances from the very first word to the very last one. This chain of queries will be completely at the user's disposal, meaning they are free to discover ranges of synonyms as far as they want. The final visual presentation could be described as a tree or network of words in which one is related to others, as crudely illustrated below.



Additional deliverables could include other useful features that show definition of a word, its various forms, the contexts it is often used in, or other words it is often associated with. These additional features are not included in the project requirements, they will be tailored according to time, but rather suggested for the sake of a more comprehensive application. Therefore, they will be considered, tailored or even rejected to ensure manageable project progress, prevent scope creep and fully leverage project resources.

Finally, regarding administrative deliverables, the source code of the project will be uploaded to a GitHub repository. Also, the final project report will include literature and data evidence to justify, validate and assess the design, implementation and value of the artefact created.

1.7. Project Significance

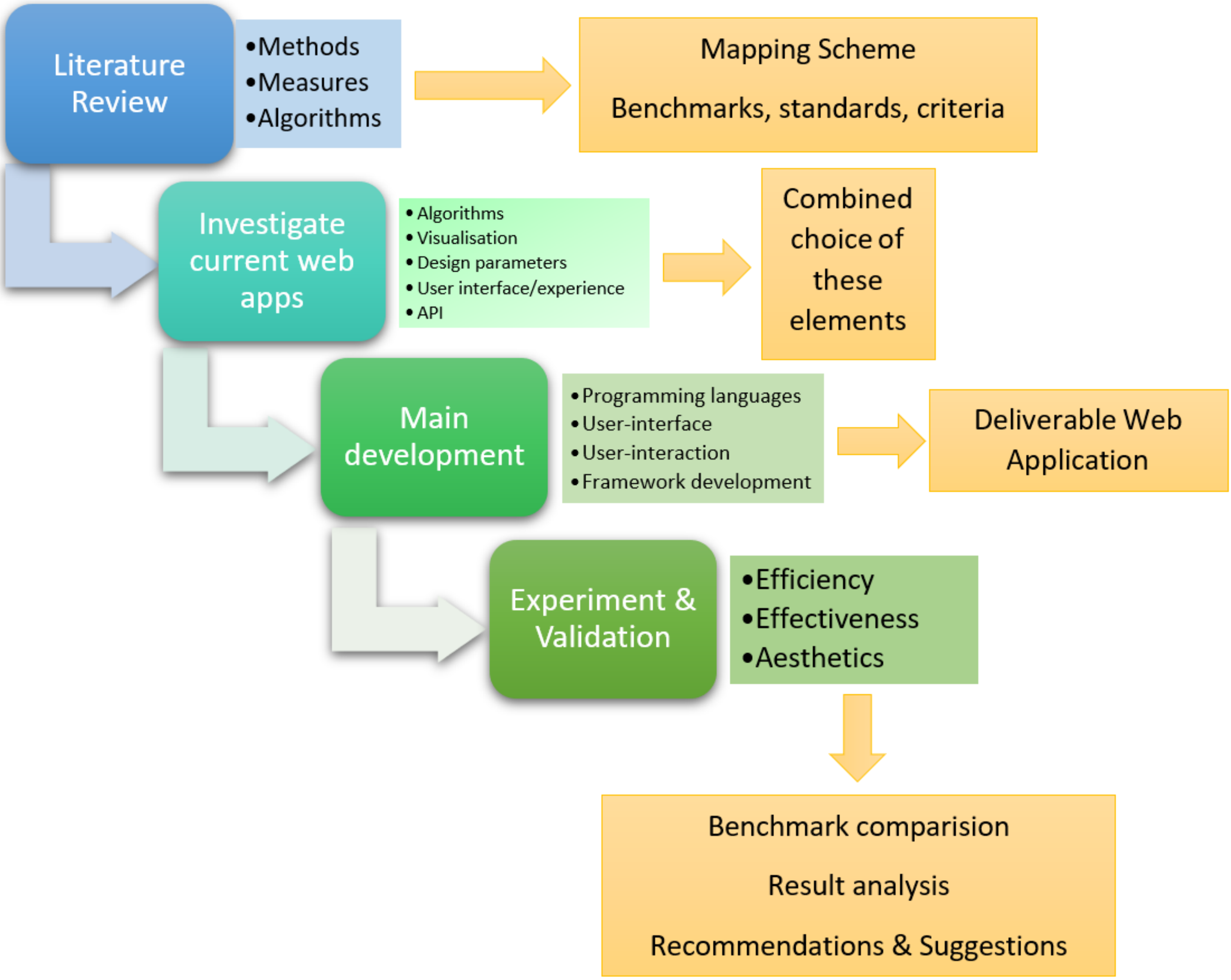
The web application to be created as the final deliverable of this project will lay the groundwork for semantic similarity quantification between two words. The application can be used not only for academic purposes, but also as a learning tool in the realm of language learning.

Furthermore, the project is expected to form the basis for extending similar functionalities to examine the closeness between phrases, sentences, paragraphs, and ultimately, documents. In this spirit, this project represents a fundamental step to a bigger project that my supervisor is looking forward to working on in the future. That ultimate project, which is now being divided into smaller and more feasible components, will seek to examine the coherence and, ultimately, the quality of a document. Even if users are not interested in using this application for academic purposes, it can still be considered simply as a learning tool in the realm of language learning.

Regarding the value of the project to research, the artefact to be created will represent an implementation of methods, algorithms or measures described in previous studies in the field. As a result, it can – to some extent – assess the validity or feasibility of these theoretical findings or concepts by applying them in a real-life application. Also, the delivered application can be utilised as a useful tool or a database for future research.

2. PROJECT METHODOLOGY

The methodology of the project is illustrated in the following flow chart which depicts briefly the stages and tasks to be completed.



A brief literature review will be conducted to understand the currently available methods, measures and algorithms in semantic similarity examination. As the application is expected to produce a score that indicates the level of similarity between two words, this background research can inform the process of choosing an appropriate scheme to map such an abstract concept as word similarity to a number that users can understand, accept and make good use of. Also, benchmarks, standards and criteria could be found in previous studies to help assess the implementation of the theoretical knowledge acquired from the literature.

The next step involves investigating the currently available web applications which generate synonyms and related words. Once the pool of current applications is created, they will be assessed more thoroughly in terms of underlying algorithms, visualisation methods, design parameters, user interface and user experience. At the same time, APIs will be searched for and extracted from online sources of dictionaries and thesauri. Each of them will then be experimented to examine its speed, feasibility, applicability and validity. At the end of this stage, decisions on a combination of suitable APIs, algorithms, visualisation, parameters and user interface will be made and applied to the proposed application.

The main development phase will involves using a variety of web application development knowledge and tools such as programming languages, user-interface, real-time interaction and framework development to build the proposed application. The goal of this process is to provide not only rapid and reliable back-end functionalities but also an intuitive and user-friendly front-end interface.

Finally, experiments will be conducted to test the efficiency, effectiveness and aesthetic appeal of the final application. Acquired results in this process can be compared against benchmarks and standards found in the previous literature review phase. An analysis of the obtained results and recommendations or suggestions for future improvement will also be delivered.

3. PROJECT MANAGEMENT APPROACH

3.1. Project Management Model

The project will be managed by ***Agile*** – a popular incremental and iterative model in planning, organising, motivating and controlling resources, procedures and protocols during a project execution (Agile project management, 2014). According to the CHAOS Manifesto (2012), IT projects employing Agile has a success rate of 42%, which is three times bigger than that of projects utilising Waterfall model. In addition, Agile has many characteristics which resonates well with the attributes of this project. They include:

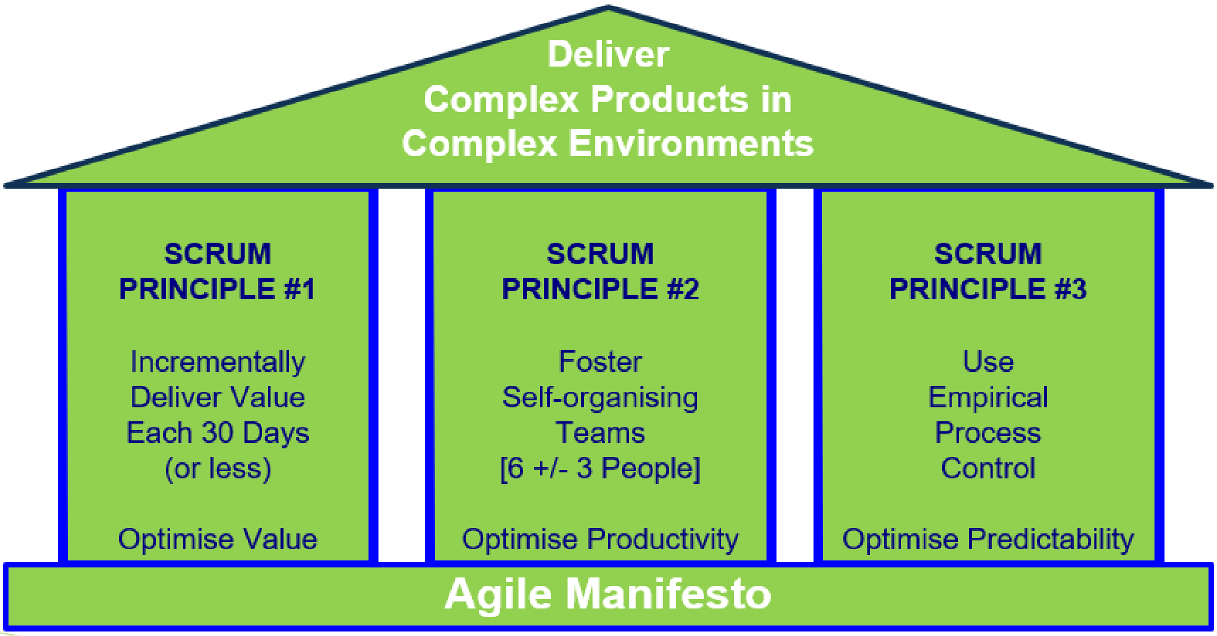
- Short-term planning scale
- Short distance between the developer and the product owner
- Short time between specification and development
- Quick to discover problems
- Low level of project schedule risks
- High ability to respond quickly to changes, especially in requirements and features of the deliverables.

Moreover, upon applying the *Cynefin framework* (Ramasesh & Browning, 2014), an IT project should belong to Complex domain as there are several associated risks described as "unknown-unknowns" that require the project team to "probe, sense and respond" (Ramasesh & Browning, 2014). The framework thus recommended Agile as the most suitable project management approach for this domain.

3.2. Project Management Framework

Scrum framework will be employed in this project as it is a simple, flexible and reliable framework for such a project of small scale. Moreover, this framework is suitable for artefact development projects as the framework considers time and cost to be fixed factors whereas features of the final deliverables are flexible (Pries & Quigley, 2010). In other words, Scrum, when applied to a project, implies that the requirements cannot be gathered upfront in a complete manner, even those successfully gathered can change, and there often seems to be more to do than time and resources can allow.

In order to optimise value, productivity and predictability of the project, I will comply strictly with three Scrum principles. In brief, I am required to be self-organising and use empirical process control to incrementally deliver value each week (Pries & Quigley, 2010).



Among these three principles, the second one captures perfectly the interaction between the student and supervisor in this project, where I am encouraged to generate ideas, implement and justify them rather than receiving specific instructions from the supervisor. The supervisor, meanwhile, as the Scrum master, keeps an eye on the project to ensure it is on the right track by providing comments, opinions and feedback. Regarding the other two principles, the first one has a close connection with the third as a burndown chart as a tool for empirical process control will be created and updated every week. This chart will show in details decreasing amount of work after each incremental period of development

4. TASK BREAKDOWN STRUCTURE AND WEEKLY PLAN

4.1. Product Backlog and MoSCoW Prioritisation

The product backlog specifies the requirements and features of the product expected to be delivered at the end of the project. Each backlog item represents a feature or functionality accompanied with an estimate of how many sprints it will take and an indicator of importance informed by *MoSCoW Prioritisation*. This mechanism of classification labels backlog item as *M – Must Have*, *S – Should Have*, *C – Could Have* or *W – Won't Have*. The specific classification below was approved by the supervisor and thus indicates both mandatory and optional components of the application required

ID	Backlog Item	Number of sprints required	MoSCoW classification
1	Synonym generator	1	M
2	Related word generator	2	M
3	Score implementation to measure similarity between words	3	M
4	Score implementation to measure similarity between phrases	3	S
5	Feature to find words following the original word	2	C
6	Feature to find concepts described by the word	2	C
7	Suggestions for misspelled input	2	C
8	Feature to find other words that describe the original word	2	C
9	Feature to find word types related to the original word	2	C
10	Feature to find forms of the word	2	C
11	Feature to find contexts the word is used in	2	C
12	Feature to find properties of the concept described by the word	2	W

13	Feature to find the use of the concept described by the word	2	W

4.2. Scrum Events

There are five events in Scrum, namely *sprint ceremony*, *sprint planning*, *daily scrum, sprint review* and *sprint retrospective*. As these events are more suitable to team development, they are adjusted to become more applicable to this individual project.

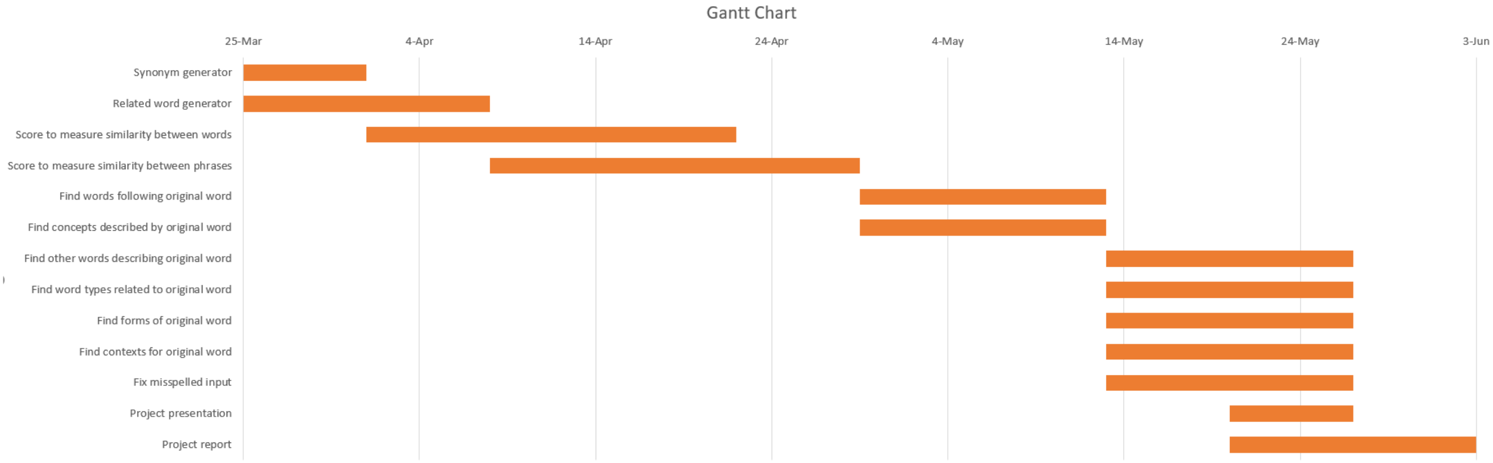
- The *sprint ceremony* is the initial meeting with the unit's teaching staff who discussed the general requirements and specifications of the project.
- The *daily scrums* executed during one week will form one *sprint*. Each daily scrum will be guided by three questions: (1) "*What did I do yesterday to meet the weekly sprint goal?*", (2) "*What am I doing today to meet that goal?*", and (3) "*Have I discovered any impediment that may prevent me from meeting that goal*?"
- The *sprint review*, *retrospective* and *planning* are cumulatively represented by the weekly meeting with the supervisor, in which the incremental value of the previous sprint is reviewed first. In detail, this review and retrospective will answer four questions, namely "*What went well?*", "*What did not?*", "*What lessons are learned?*", and "*What to do differently?*". As a result, decisions on whether to continue, stop or initiate some action will be made.
- Afterwards, in the same meeting, the *sprint planning* event will commence with the next sprint backlog items being discussed and set as the goals for the next sprint. During each meeting, the product backlog, product breakdown structure, work breakdown structure, product flow diagram and burndown chart will be updated to clearly demonstrate the overall progress. Also, comments, opinions and feedback from the supervisor will be collected in order to make any necessary adjustments in a timely and efficient manner.

4.3. Task Breakdown Structure and Weekly Plan

The task breakdown structure below provides a detailed timeline for the project. According to recommendation from the supervisor, the *Must-Haves* and *Should-Haves* determined in section 4.1 are prioritised to be developed and polished within this semester, while the *Could-Haves* and *Won't Haves* should be left for a possible subsequent project in the next semester. Therefore, the execution plan below only includes the Must-Haves and Could-Haves in order to deliver the proposed application adequately, optimally and satisfactorily.

Time period	Activity	Deliverables
Project setup and preparation		
Week 1	Initial meeting with supervisor	Project approval
Week 2	• Initial research into the topic • Complete project agreement	Project agreement
Week 3	• Draft project schedule and plan • Complete project pitching/presentation • Gather comments and feedback from presentation examiner	Project presentation
Week 4	Complete project plan	Project plan
Development of Must-Haves and Should-Haves (Sprint 1 - 6)		
Sprint 1 **Week 5**	Conduct detailed literature review	• Approaches, methods, algorithms and measures • Benchmarks, standards and criteria for implementation
Sprint 2 **Week 6**	Investigate current applications, online dictionaries, thesauri	Proposals of: • Algorithm • API • Visualisation method • Design parameters • UI/UX
Sprint 3 **Week 7**	Develop feature for synonym queries	Module for synonym queries
Sprint 4 **Week 8**	Develop feature for related word queries	Module for related word queries
Sprint 5-7 **Week 9-11**	Implement score feature to measure similarity between and (optional) phrases	Score feature to quantify similarity between words and (optional) phrases
Product Deployment and Project Closure		
Sprint 8 **Week 12**	• Deploy/Launch application • Test benchmarks	Final web application
Sprint 9 **Week 13**	• Complete project report	Project report

The weekly plan above is visually demonstrated in the Gantt chart below



	Project report	Project presentation	Fix misspelled input	Find contexts for original word	Find forms of original word	Find word types related to original word	Find other words describing original word	Find concepts described by original word	Find words following original word	Score to measure similarity between phrases	Score to measure similarity between words	Related word generator	Synonym generator
Start Date	20-May	20-May	13-May	13-May	13-May	13-May	13-May	29-Apr	29-Apr	8-Apr	1-Apr	25-Mar	25-Mar
■ Days to complete	14	7	14	14	14	14	14	14	14	21	21	14	7

5. COMMUNICATION PLAN

In order to successfully execute the project plan, the student and supervisor will continuously monitor the project and ensure it is on the right track through close and frequent communication. The communication plan is presented in the table below. The participants for almost all communication items are the student and supervisor, with the exception of weekly progress update additionally including the project coordinator.

Item	Objective	Method	Frequency	Deliverable
Sprint planning	• Discuss next sprint backlog items to be developed • Set goals for the next sprint.	Face to face	Weekly Tuesday 12 pm	• Product backlog • Product breakdown structure
Sprint Review and Retrospective	• Review incremental value of previous sprint • Gather comments and feedback • Decide whether to continue, stop or initiate some action	Face to face	Weekly Tuesday 12 pm	• Product flow diagram • Work breakdown structure • Incremental product component
Project progress update	Inform the project coordinator and	Email	Weekly	• Product flow diagram • Weekly journal

	supervisor of the project progress			• Log sheet
General concerns or enquiries	Discuss any concerns, ideas or enquiries	Email Face to face	Frequent As required	
Consolidation review	Review significant chunks of developed components	Face to face	End of Week 8, 10 and 12	• Product Backlog • Product Flow Diagram • Product breakdown structure

6. RISK ASSESSMENT

This section specifies the risks associated with the project as well as their mitigation and contingency plans. A risk mitigation plan requires its proposed actions to be taken in advance irrespective of the impact or probability of the risk. This practice aims to prevent risks before they happen and reduce their impact and probability (Kendrick, 2009). Meanwhile, a risk contingency plan represents a set of fallback action after a risk has happened in order to control the impact, especially in case the mitigation plan has not been effective (Kendrick, 2009). In other words, action specified in a risk contingency plan does not need to be taken immediately, but certain warning signs should be monitored closely should resort to the contingency plan be required.

ID	Description	Probability	Impact
1	Insufficient time	Medium	High
2	Unavailability of supervisor	Low	Medium
3	Lack of technical skills and knowledge	Medium	High
4	Scope creep	Low	High
5	Limited access to online repositories, resources or database	Low	Medium
6	Delivery of wrong solution, product or application	Low	High
7	Implementation not corresponding with methods in literature	Medium	Medium
8	Low quality of solution	Low	Medium

Description	Mitigation plan	Contingency plan
Insufficient time	• Closely monitor progress each week • Clear project plan and follow it strictly • Discuss any impediment immediately	Exclude unnecessary features

Unavailability of supervisor	• Ensure clear meeting schedule • Frequently communicate to timely inform unavailability	• Seek help from the unit teaching staff • Plan more meetings in advance
Lack of technical skills and knowledge	• Proactively learn new skills and knowledge • Persevere in seeking solutions from online learning sources (forums, blogs, etc)	Seek help from QUT Initiatives (STIMulate, Student Success Group)
Scope creep	• Agree on deliverables in advance • Use Agile and Scrum to prioritise features to be delivered	• Justify that the current scope is rational • Assess if remaining time is sufficient for changes
Limited access to online repositories, resources or database	Gather beforehand as many resources as possible	• Seek help from the supervisor in requesting access • Pay for access
Delivery of wrong solution, product or application	Deliver incremental component frequently to keep the project on the right track	None as this is fatal. Avoid at all costs
Implementation not corresponding with methods in literature	Discuss with supervisor and seek approval before implementation	Reconsider/rebuild the implementation
Low quality of solution	Consistently seek feedback from supervisor	• Reconsider/rebuild solution • Take as lesson learned

7. PROJECT CONSTRAINTS

The project is characterised by three standard contraints, namely resources, time and scope. Firstly, resource constraints refer to the specific level of knowledge and skills in programming and application development of the student. In addition, lack of good equipment could represent a minor contraint in which the technology infrastructure such as computers or servers supplied by the student may not be powerful to facilitate fast functionalities and achieve accurate benchmarks. Secondly, time is a significant contraint which requires the application to be fully developed and delivered within eight weeks. Finally, due to the novelty of the research problem, the project scope has to be limited to the development of the most fundamental and essential functionalities for the application. Consequently, other features that may make the application more versatile and comprehensive are left out.

REFERENCES

Agile project management. (2014). Ashford, Kent: DSDM Corporation.

CHAOS Manifesto. (2012). *The Standish Group International*. Retrieved from https://cs.calvin.edu/courses/cs/262/kvlinden/resources/CHAOSManifesto2012.pdf

Chaudhuri, S., Ganti, V., & Kaushik, R. (2006). A Primitive Operator for Similarity Joins in Data Cleaning. In *22nd International Conference on Data Engineering*. https://doi.org/10.1109/ICDE.2006.9

Kendrick, T. (2009). *Identifying and managing project risk essential tools for failure-proofing your project* (2nd ed.). New York: AMACON.

Kondrak, G. (2005). N-gram similarity and distance. *Lecture Notes in Computer Science, 3772*, 115–126. https://doi.org/10.1007/11575832_13

Oliva, J., Serrano, J., Del Castillo, M., & Iglesias, Á. (2011). SyMSS: A syntax-based measure for short-text semantic similarity. *Data & Knowledge Engineering*, *70*(4), 390–405. https://doi.org/10.1016/j.datak.2011.01.002

Pries, K., & Quigley, J. (2010). *Scrum project management*. Boca Raton, FL: CRC Press.

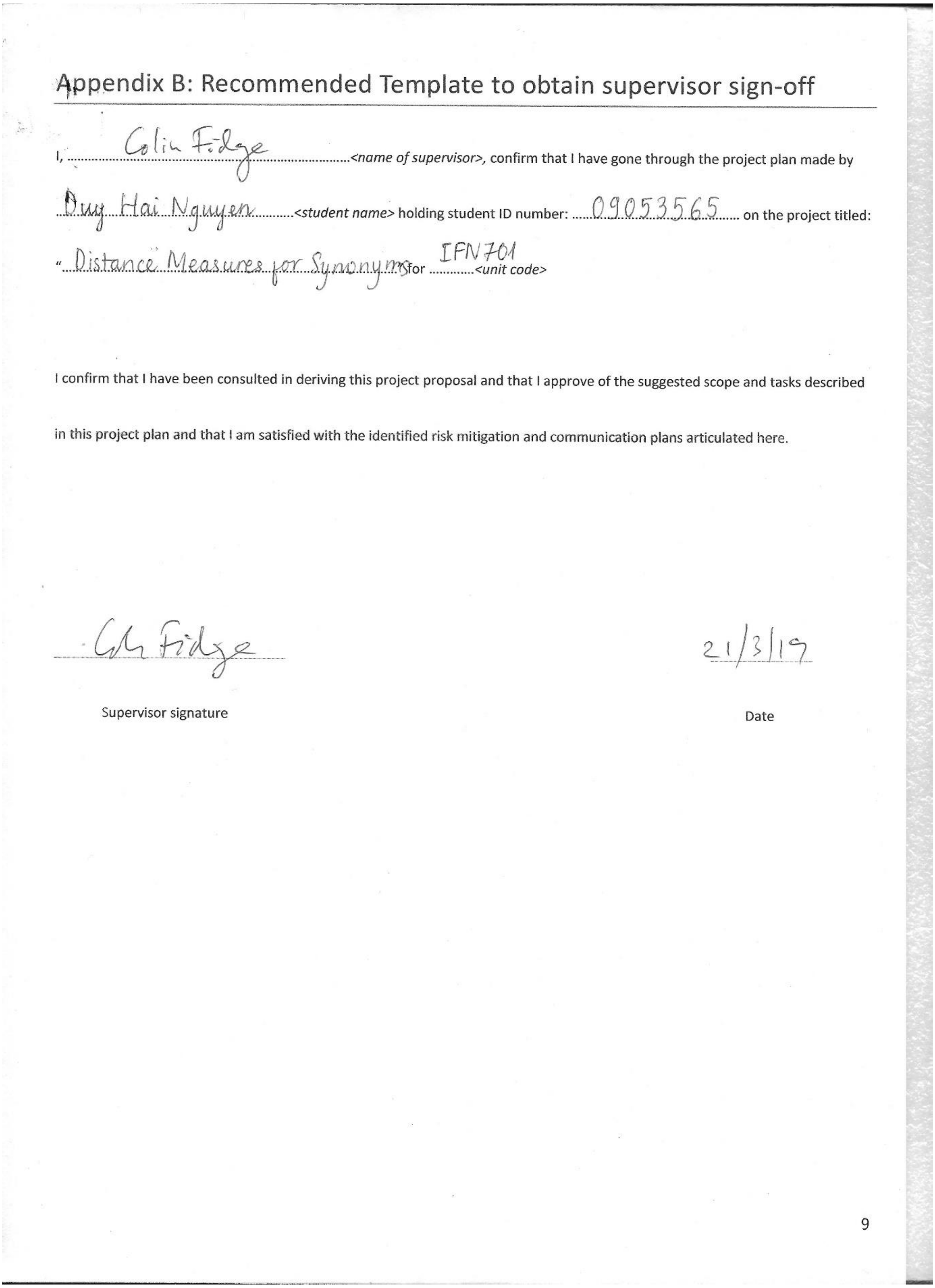
Ramasesh, R., & Browning, T. (2014). A conceptual framework for tackling knowable unknown unknowns in project management. *Journal of Operations Management, 32*(4), 190–204. https://doi.org/10.1016/j.jom.2014.03.003

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management, 24*(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

Schepens, J., Dijkstra, T., & Grootjen, F. (2012). Distributions of Cognates in Europe as Based on Levenshtein Distance. *Bilingualism: Language and Cognition*, *15*(1), 157–166. https://doi.org/10.1017/S1366728910000623

Wang, Y., Qin, J., & Wang, W. (2017). Efficient approximate entity matching using Jaro-Winkler distance. *Lecture Notes in Computer Scienc, 10569*, 231–239. Springer Verlag. https://doi.org/10.1007/978-3-319-68783-4_16

SUPERVISOR’S APPROVAL



Appendix B: Recommended Template to obtain supervisor sign-off

I, ...Colin Fidge......<*name of supervisor*>, confirm that I have gone through the project plan made by

....Duy Hai Nguyen....<*student name*> holding student ID number: ....09053565.... on the project titled:

“....Distance Measures for Synonyms....” for ....IFN701....<*unit code*>

I confirm that I have been consulted in deriving this project proposal and that I approve of the suggested scope and tasks described in this project plan and that I am satisfied with the identified risk mitigation and communication plans articulated here.

Colin Fidge

Supervisor signature

21/3/19

Date