Project 1

Mohammed Musthafa Rafi
Computer Science, PhD, Iowa State University
mohd7@iastate.edu

Abstract

This report summarizes a text analysis pipeline for animal QTL-related papers. The tasks included text pre-processing, creating word clouds (via word frequency and TF-IDF), training Word2Vec models on both word- and phrase-level corpora, and examining how many extracted phrases matched a domain-specific trait dictionary. The project also explored an alternative approach to phrase extraction using Gensim `Phrases`.

1 Method

Data and Pre-processing. Abstracts labeled `Category=1` were retained. Each abstract was split into sentences, tokenized, lowercased, and stripped of stop words and punctuation. Numeric tokens and short tokens (length ≤ 1) were removed.

Task 1: Word Cloud Construction. Two word clouds were generated using:

- **Frequency-based**: Raw word counts.
- **TF-IDF-based**: Term frequencies weighted by inverse document frequency.

Task 2: Word2Vec Model. A Word2Vec model was trained with `vector_size=100`, `window=5`, `min_count=10`. The 20 most similar words were retrieved for each of the top 10 TF-IDF words.

Task 3: Phrase Extraction. Bigrams and trigrams were initially extracted by simple n-grams, and a new Word2Vec model was trained on these phrase-level tokens. Two additional word clouds (frequency and TF-IDF) were generated from phrases, and the 20 most similar phrases were listed for the top 10 frequent trait-related phrases.

2 Main Results

2.1 Word Cloud Observations

Figures 1 and 2 show that raw frequency highlights common words like *qtl*, *gene*, *traits*, while TF-IDF emphasizes more specialized terms such as *snps* and *genome*.

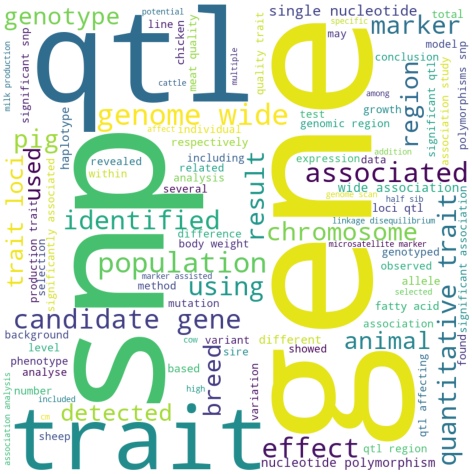

Figure 1: Frequency-based Word Cloud (words).

2.2 Word2Vec Observations

For top TF-IDF words (*qtl*, *traits*, *snps*, etc.), similar words generally relate to QTL or genetics. Positive examples include *qtl* $\rightarrow$ {*qtls*, *locus*, *mapped*} and *snps* $\rightarrow$ {*snp*, *polymorphism*, *haplotypes*}. Some terms appeared less relevant, likely due to corpus noise.

2.3 Phrase-Level Word Clouds and Similarities

Task 3 produced word clouds at the phrase level (Figures 3–4). Frequent bigrams and trigrams like *body weight* or *fatty acid composition* emerged clearly. Similarity queries for these phrases showed neighbors like *body weight* $\rightarrow$ {*average_daily_gain*, *length*}.

2.4 Dictionary Matching

Out of 291,437 extracted phrases, 5559 matched the Trait Dictionary. Common matches included *body weight* and *litter size*.

Figure 2: TF-IDF-based Word Cloud (words).

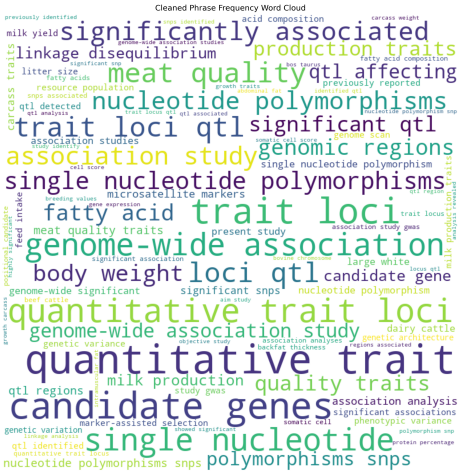

Figure 3: Frequency-based Word Cloud (phrases).

3 Discussion

An alternative approach to phrase extraction was tested with Gensim `Phrases`, which automatically detects multi-word expressions (e.g., *body_weight*). This method found 168 unique trait-related phrases with default settings, versus 5 from the simpler n-gram approach. With more lenient parameters (`min_count=3, threshold=5.0`), 198 phrases emerged, but introduced additional noise.

Expectations vs. Reality. It was expected that lower thresholds would reveal more relevant domain-specific phrases. While additional phrases (like *chest circumference*, *color l*, *cell differentiation*) were discovered, many were contextually less relevant. Thus, a balance between capturing domain-specific n-grams and maintaining precision is needed.

Limitations. Only abstracts labeled `Category=1` were used, and simple heuristics were applied for pre-processing. Larger, more diverse corpora or advanced filtering might improve precision, and domain-specific phrase dictionaries could refine results.

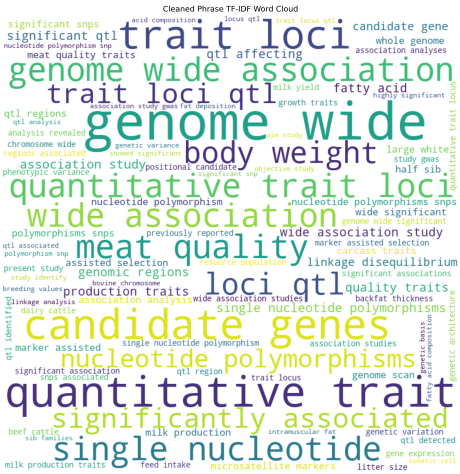

Figure 4: TF-IDF-based Word Cloud (phrases).

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

A Appendix: Resources

Below are links to tutorials and external references:

- https://github.com/itsMustafamr/qtl_text_analysis_project_579 (GitHub repository containing code, data processing scripts, and README instructions)
- https://github.com/amueller/word_cloud (WordCloud in Python)
- https://radimrehurek.com/gensim/ (Gensim Word2Vec and Phrases)
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
- https://www.nltk.org/ (NLTK for tokenization)
- https://www.youtube.com/watch?v=--netCHZyL4 (Sample tutorial on Word2Vec)

All code and instructions for running the analysis, including preprocessing steps and model training, are available in the GitHub repository.

Code was primarily written in Python, using packages `gensim`, `nltk`, and `scikit-learn`