Natural Language Processing with Deep Learning Technology

for

Yelp Review Polarity Analysis

Version 1.0 approved

Prepared by Joonhee Lee, Young-O Won, Sangjin Kang

KAIST College of Business

2022-05-27

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Joonhee Lee | 05-27 | Initiate (Intro and Blueprint) | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

소비자 온라인 리뷰는 소비자의 의사 결정 과정에 필수적인 부분이 되었습니다. 최근 연구에 따르면 온라인 리뷰는 93% 소비자들의 구매 결정에 영향을 미치며 (Kaimingk 2019), 91%의 소비자가 온라인 리뷰를 개인적인 추천만큼 신뢰하는 것으로 나타났습니다 (Igniyte 2019). 온라인 리뷰는 비즈니스에 실체적인 경제적 영향을 미치는 것으로 나타났습니다 (Moe and Trusov 2011).

경영학 학계에서의 근래까지의 연구는 별점과 같은 numerical data 및 형식화된 데이터를 기반으로 하는데, 온라인 리뷰는 일반적으로 별점과 같은 numerical evaluation과 textual evaluation으로 구성된다는 것을 고려할 때, textual evaluation이라는 최근 더 부각되는 리뷰의 비중을 고려하지 못하고 있다는 한계가 있습니다 (Schoenmueller, Netzer, and Stahl, 2020). 공학계에서는 자연어처리와 같은 딥러닝 기반 기술을 통해 classification 연구가 많이 진행되고 있지만, 비즈니스와의 연계성은 부족하다고 생각합니다. 저희 팀이 찾은 가장 비즈니스와 유력한 연계성을 갖는 연구 중 하나는 판매 예측에 대한 리뷰 textual evaluation의 관련성을 보여주는 연구였습니다(Archak, Ghose and Ipeirotis 2011). 저희는 텍스트 평가의 분석이 중요하다고 믿으며, 검토 텍스트의 기본 감정을 조사하고 numerical evaluation 분포와 textual evaluation의 잠재적 차이를 식별하기 위해 향후 연구를 장려해야 한다는 혹자의 의견이 존재합니다.

저희 팀의 이번 강의 Term Project의 목표로 가장 frontier의 자연어처리 등을 비롯한 딥러닝 기반의 기술을 통해 온라인 리뷰의 textual evaluations들에 대한 분석을 진행하는 것 입니다. 이를 기반으로 새로운 리뷰들에 대한 classification도 성공적으로 진행하는 것을 목표로 해보겠습니다.

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## References

Archak, Nikolay, Anindya Ghose, and Panagiotis G. Ipeirotis (2011), “Deriving the Pricing Power of Product Features by Mining Consumer Reviews,” Management Science, 57 (8), 1485–1509.

Igniyte (2019), “30 Essential Online Review Facts and Stats,” (October 29), <https://www.igniyte.co.uk/blog/30-online-review-factsand-stats/>.

Kaemingk, Diana (2019), “20 Online Review Stats to Know in 2019,” Qualtrics (April 9), <https://www.qualtrics.com/blog/onlinereview-stats/>.

Moe, Wendy W. and Michael Trusov (2011), “The Value of Social Dynamics in Online Product Ratings Forums,” Journal of Marketing Research, 48 (3), 444–56.

Schoenmueller V, Netzer O, Stahl F. The Polarity of Online Reviews: Prevalence, Drivers and Implications. Journal of Marketing Research. 2020, 57(5), 853-877.

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

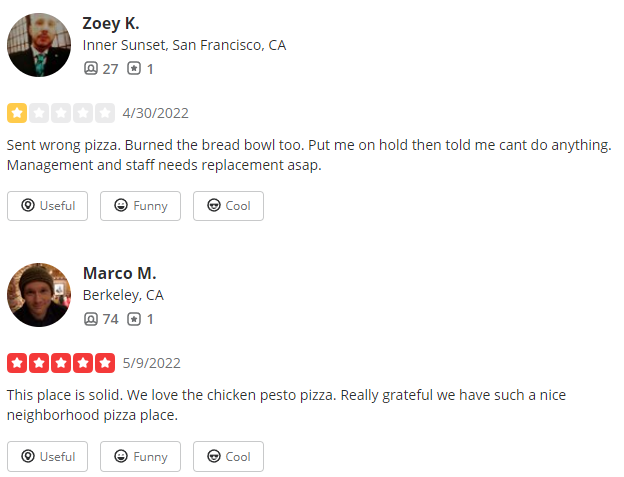
## Product Functions

본 프로젝트의 목표는 우리가 얼마나 잘 지난 Yelp 플랫폼의 리뷰들을 알고리즘을 훈련시킴으로써 새로운 Yelp Reivew들을 분류할 수 있는지 알아보는 것입니다.

알고리즘에 기반이 되는 중추적인 기술은 딥러닝과 그 기반의 자연어 처리가 될 것이라 예상합니다. 1990년대 이후에는 대량의 말뭉치(corpus) 데이터를 활용하는 기계학습 기반 및 통계적 자연어 처리 기법이 주류를 이뤘으나, 최근에는 딥러닝과 딥러닝 기반의 자연어처리가 방대한 텍스트로부터 의미 있는 정보를 추출하고 활용하기 위한 언어처리 연구 개발이 전 세계적으로 활발히 진행되고 있습니다.

자연어처리 기술은 기계번역, 대화체 질의응답 시스템 대화시스템, 정보검색, 말뭉치 구축, 시맨틱 웹, 텍, 딥러닝, 그리고 빅데이터 분석 분야 뿐만 아니라 인간의 언어정보처리 원리와 이해를 위한 언어학과 뇌인지 언어정보처리 분야까지 핵심적인 요소로 작용하고 있습니다.

기본적인 청사진은 training data로 Yelp 플랫폼의 리뷰를 활용하는 것입니다. 아래의 사진과 같이 Yelp 상의 리뷰들은 기본적으로 별점과 같은 numerical rating 뿐만 아니라, textual evaluation 또한 일반적으로 병행됩니다. 우리는 이러한 특징을 활용해서 textual evaluation을 기반으로 numerical rating을 예상할 수 있고, 나아가 recommendation의 정도를 추정해볼 수 있습니다. 또한, mid-range (별점 3점) 리뷰들이 실제로는 어떤 특징을 가지고 있는지 구현되는 모델을 통해 나열해볼 수 있을 것으로 사료됩니다.



## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

[Using FastAI to Analyze Yelp Reviews and Predict User Ratings (Polarity) | by Sho Fola | DataDrivenInvestor](https://medium.datadriveninvestor.com/using-fastai-to-analyze-yelp-reviews-and-predict-user-ratings-polarity-4e4e89df358e)

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements