

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Informatica

A.A. 2024/2025

Sessione di Laurea Aprile 2025

**INTERAZIONI ADATTIVE TRA BOT E
UTENTI: LINEE GUIDA PER LA
PROGETTAZIONE DI SISTEMI DINAMICI E
COERENTI**

Relatore:

prof. Marco Mazzaglia

Candidato:

Daniele Femia

INDICE

1. INTRODUZIONE	4
1.1 Contesto e Obiettivi della ricerca	4
1.2 Contesti di utilizzo.....	5
2. STATO DELL'ARTE	6
2.1 Storia degli NPC.....	6
1970s: NPC estremamente rigidi e scriptati	6
1980s: Miglioramenti nei movimenti e nelle linee di dialogo.....	6
1990s: Algoritmi goal-oriented	7
2000s: Statistical Learning e routine avanzate	7
2010s: Machine Learning e reti neurali.....	8
2.2 Futuro degli NPC: l'era degli LLM.....	9
NVIDIA ACE: NPC con intelligenza generativa	9
Inworld AI: creare NPC con personalità via LLM	10
2.3 Ricerche Accademiche e Sviluppi sugli NPC basati su LLM	11
What if Red Can Talk? Dynamic Dialogue Generation Using Large Language Models.....	11
Leveraging Pre-trained Language Models for Interactive NPC Dialogue Generation.....	11
Memory Repository for AI NPC	12
3. IDEAZIONE DEL SISTEMA.....	13
3.1. Concetti di base	13
3.1.1. Spiegazione e origine dell'idea	13
La Dinamica delle Mappe Concettuali	13
Astrazione in Gruppi e Propagazione Inter-Gruppo.....	14
Il Ruolo degli Eventi nella Propagazione delle Informazioni.....	15
Concetto di Pensiero sul Giocatore e sui Gruppi.....	15
I Pilastri del Sistema e i Benefici del Modello	16
3.1.2. Introduzione delle fondamenta: eventi, mappa concettuale e pensiero sul giocatore e sul gruppo	17
Gli Eventi.....	17
La Mappa Concettuale.....	17
Il Pensiero sul Giocatore.....	18
Il Pensiero Sugli Altri Gruppi.....	18
3.2 Generazione di un nuovo evento	19
Descrizione dell'Evento	19
Tipologia e Importanza.....	19
Generazione delle Statistiche.....	20
3.3. Diffusione degli eventi: prima idea	21
Approccio Deterministico vs. Probabilistico.....	21
Propagazione degli Eventi all'Interno del Gruppo.....	21

Propagazione degli Eventi all'Esterno del Gruppo	22
Integrazione dei Concetti Temporali	23
Unit.....	24
Problemi con Questa Soluzione.....	25
3.4 Diffusione degli eventi: seconda idea.....	27
Propagazione degli Eventi all'Interno del Gruppo e Velocità	27
Velocità di Diffusione Interna	29
Diffusione degli Eventi all'Esterno del Gruppo e Velocità	29
I Link tra i Gruppi: Nuovo Utilizzo.....	30
Osservazioni Finali.....	31
3.5 Gli NPC: Descrizione e Interazione con gli LLM.....	32
3.5.1 Definizione e Strutturazione degli NPC	32
Aggiornamento del Pensiero sul Giocatore e sul Gruppo	33
Pensiero sul Gruppo	34
File di Configurazione per il Comportamento degli NPC	34
3.5.2 Creazione di Nuovi NPC	37
Generazione degli Attributi	37
Calcolo degli Attributi.....	37
3.5.3 Comportamento degli NPC alla Ricezione degli Eventi	40
Fase di Filtraggio dell'Evento.....	40
Aggiornamento delle relazioni	40
Distorsione dell'Evento.....	41
Trigger dell'Aggiornamento dei Pensieri.....	42
Controllo della Similarità tra Eventi.....	42
Propagazione dell'Evento	43
3.5.4. Interazione con gli LLM.....	44
RTGS System.....	47
Composizione del Prompt	50
3.5.5 Gestione della Memoria.....	51
Registro della Memoria	52
Sintesi della Memoria.....	52
Recupero della Memoria	53
Analisi Comparativa con il paper Memory Repository for AI NPC	53
Integrazione dei Concetti e Flusso Operativo.....	54
3.6 Gruppi.....	55
Generazione di un Evento Casuale tra NPC.....	55
Gestione del Pensiero Comune sul Giocatore e Sugli Altri Gruppi	56
Aggiornamento dei Link in Base al Pensiero Comune.....	56

Gerarchia	58
4. CASI D'USO	59
4.1. Esempio di utilizzo del sistema	60
Introduzione dell'Esempio e Obiettivo	60
Contesto dell'Esempio: Il Continente del 1400	60
Com'è composto l'esempio	61
4.2. Esempio fuori dai videogiochi.....	62
Introduzione dell'Esempio e Obiettivo	62
Contesto dell'Esempio: Consegna del Progetto	62
Com'è composto l'esempio	63
4.3. Esempio assistente di gioco.....	64
Introduzione dell'Esempio e Obiettivo	64
Contesto dell'Esempio: Il Gioco da Tavola e il Ruolo dell'Assistente	64
Risultati Attesi e Valutazione delle Dinamiche.....	64
Com'è composto l'esempio	65
5. I Risultati	66
5.1. Valutazione della diffusione degli eventi	66
Probabilità di Diffusione e Velocità Interna.....	66
Probabilità di Diffusione e Velocità Esterna	68
Aggiornamento dei Link tra NPC.....	69
Aggiornamento dei Link tra Gruppi con Propagazione a Catena.....	70
5.2. Dinamiche Interattive e Risposte Contestuali degli NPC.....	72
5.3. Valutazione del Sistema RTGS	76
5.4. Valutazione della Componente della Memoria	80
Caso 1: Recupero dalla Memoria a Breve Termine.....	81
Caso 2: Recupero dalla Memoria a Lungo Termine.....	82
Caso 3: Gestione di Molteplici Ricordi e Selezione tramite Cosine Similarity.....	82
Caso 4: Transizione dalla Memoria a Breve Termine a quella a Lungo Termine.....	82
6. Linee Guida	83
Integrazione Dinamica delle Relazioni.....	83
Impatto Contestuale delle Conversazioni sull'NPC.....	83
Propagazione degli Eventi.....	84
Flessibilità Parametrica e Adattabilità Contestuale	85
7. CONCLUSIONI.....	86
7.1 Riassunto	86
7.2 Miglioramenti futuri	87
Bibliografia.....	88

1. INTRODUZIONE

1.1 Contesto e Obiettivi della ricerca

Negli ultimi anni l'intelligenza artificiale ha avuto uno sviluppo eccezionale, soprattutto per quanto riguarda l'evoluzione dei Large Language Model (LLM). Questi modelli, che sfruttano architetture di deep learning, hanno rivoluzionato il modo in cui l'uomo e le macchine interagiscono. Comprendono e generano il linguaggio naturale, permettendo una conversazione fluida e contestuale, cosa che, fino a poco tempo fa, sembrava irraggiungibile. La capacità incredibile degli LLM di imparare da vasti dataset e di sintetizzare informazioni complesse ha aperto nuove prospettive per la realizzazione di agenti conversazionali, rendendo possibile la creazione di sistemi in cui l'utente può interagire liberamente e naturalmente. Questo sviluppo rappresenta non solo un salto qualitativo nella capacità computazionale, ma anche un cambio di paradigma nell'interazione uomo-macchina, dove la flessibilità e l'adattabilità del linguaggio giocano un ruolo centrale.

Nel contesto videoludico e delle simulazioni interattive, il concetto di Non Player Characters (NPC) è da sempre al centro degli sforzi degli sviluppatori, che mirano a garantire interazioni coerenti e coinvolgenti per l'utente. Nei sistemi tradizionali, gli NPC sono generalmente progettati per agire seguendo delle regole predefinite che consentono di propagare le scelte del giocatore a una rete di personaggi interconnessi, rendendo l'esperienza narrativa dinamica e interattiva. Questi sistemi, sebbene efficaci nel creare una coerenza interna alla narrazione, presentano delle limitazioni intrinseche, in quanto tendono a operare secondo meccanismi rigidamente strutturati, in cui ogni decisione dell'utente ha effetti propagati in maniera deterministica all'interno dell'ecosistema di gioco.

Con l'avvento degli LLM, si è aperta la possibilità di alimentare gli NPC con capacità conversazionali estremamente sofisticate, capaci di rispondere in tempo reale a domande, adattarsi a contesti mutevoli e sostenere conversazioni aperte e libere. Uno dei principali problemi riscontrati in questo nuovo approccio è la mancanza di una propagazione delle influenze che vada oltre il semplice scambio diretto con l'utente. Mentre i sistemi classici consentono che una scelta del giocatore influenzi indirettamente altri NPC, creando una rete interattiva in cui le azioni si diffondono e si amplificano, gli NPC alimentati da LLM spesso operano come entità isolate. Sebbene un LLM possa generare risposte coerenti e contestuali durante una conversazione, il modello attuale non integra meccanismi per trasmettere l'impatto di quella conversazione ad altri agenti all'interno dello stesso ambiente. Questo gap rappresenta una delle carenze principali delle soluzioni basate su LLM: il potenziale delle interazioni si limita a essere locale, senza una vera propagazione delle decisioni e delle scelte del giocatore a livello sistemico.

Il problema fondamentale risiede nella capacità degli NPC alimentati da LLM di essere influenzati in maniera globale. Le interazioni, pur essendo di elevata qualità in termini di fluidità e coerenza linguistica, non si traducono in un impatto diffuso sul sistema narrativo. La mancanza di un meccanismo che permetta di propagare le azioni del giocatore ad una molteplicità di agenti porta a una frattura nell'esperienza interattiva: il dialogo con un singolo NPC non contribuisce a modificare le interazioni successive o lo stato dell'intero ecosistema, ma rimane confinato a quella specifica interazione. Questo limita fortemente l'immersione e la dinamicità della narrazione, dato che il mondo simulato non si evolve in maniera organica in risposta alle scelte dell'utente.

L'obiettivo di questo lavoro è quello di creare delle linee guida che aiutano nella creazione di sistemi (gettare le basi per un nuovo paradigma di interazione) in cui gli NPC alimentati da LLM possano ricevere e propagare le influenze in maniera diffusa. L'idea centrale è quella di trasformare ogni interazione in un evento che abbia il potenziale di influenzare indirettamente altri agenti. In questo modo si mira a creare un sistema in cui le scelte dell'utente non siano considerate in maniera isolata, ma si integrino in una rete interconnessa di influenze che rende l'esperienza narrativa più ricca e complessa.

1.2 Contesti di utilizzo

Il sistema proposto mira ad essere utilizzato in una vasta gamma di contesti, andando oltre l'ambito videoludico e aprendo nuove prospettive per applicazioni in ambiti come le simulazioni ambientali, la formazione interattiva, gli assistenti virtuali e altri settori in cui la dinamica delle interazioni risulta fondamentale.

Nel mondo dei videogiochi, ad esempio, questo sistema permette di creare ambienti narrativi interconnessi, in cui le decisioni dell'utente non sono tra semplici scelte multiple, ma azioni libere che hanno impatti significativi e che si propagano all'interno del mondo narrativo. Questo significa che ogni decisione o azione fatta dall'utente contribuisce ad influenzare l'evoluzione del mondo virtuale, rendendo l'esperienza di gioco più immersiva e realistica consentendo la modellazione di una esperienza dell'utente estremamente ricca e personalizzata.

Oltre al settore videoludico, il sistema può essere facilmente adattato in contesti come le simulazioni ambientali. Qui, la capacità di propagare eventi e influenze in tempo reale permette di riprodurre scenari complessi, in cui le azioni di un singolo elemento possono avere effetti a catena su un intero ecosistema. In un contesto di questo tipo, la simulazione diventa uno strumento per comprendere le dinamiche naturali che regolano i sistemi complessi, facilitando l'identificazione di processi connessi tra di loro e la previsione di eventuali effetti a catena dovuti ad interventi mirati.

Il settore della formazione interattiva rappresenta un campo in cui il sistema si dimostra particolarmente efficace. La possibilità di creare ambienti virtuali dinamici e interattivi, nei quali le scelte dell'utente si traducono in conseguenze, offre un sistema ideale dove gli utenti possono imparare a prendere delle decisioni affrontando direttamente le conseguenze di esse e migliorare quindi le competenze decisionali e di *problem solving*. La libertà di azione, non vincolata da opzioni preimpostate, consente di affrontare situazioni complesse in modo del tutto personalizzato, migliorando così l'efficacia del processo formativo.

In ambito assistenza virtuale, il sistema consente una propagazione dinamica degli eventi tramite una rete interconnessa di agenti. Tale meccanismo permette all'assistente di adattare il proprio comportamento in tempo reale, integrando informazioni da interazioni precedenti per offrire risposte contestuali e personalizzate. L'utente, non vincolato a scelte predefinite, beneficia di un'interazione flessibile e naturale, che regola tono e contenuto del dialogo in funzione del contesto. Questo approccio innovativo migliora la gestione delle dinamiche comunicative e favorisce lo sviluppo di relazioni di fiducia.

Il principale vantaggio di questo sistema deriva proprio dalla capacità di modificare il modo con cui gli agenti rispondono in base alle conversazioni precedenti e quelle degli altri agenti. In un ambiente dove ogni NPC o assistente virtuale si comporta in modo indipendente, l'interazione risulta frammentata e l'impatto delle scelte dell'utente si limita al singolo contatto. Al contrario, grazie alla rete interconnessa di agenti, il sistema permette di realizzare una propagazione degli effetti delle interazioni, rendendo possibile una comunicazione che si estende ben oltre il semplice scambio diretto. Questo approccio favorisce la creazione di un ecosistema in cui le influenze si moltiplicano e si intrecciano, portando a una narrazione globale e coesa che arricchisce l'esperienza utente.

2. STATO DELL'ARTE

2.1 Storia degli NPC

1970s: NPC estremamente rigidi e scriptati

Negli anni '70 gli NPC avevano comportamenti semplici e totalmente predefiniti, a causa dei forti limiti hardware dell'epoca. Nei primi arcade con modalità giocatore singolo, i "nemici" controllati dal computer seguivano schemi fissi di movimento e attacco, senza vera adattabilità. Ad esempio, **Space Invaders** (1978) presentava ondate di alieni che si muovevano in formazione con pattern determinati a priori, semplicemente accelerando man mano che venivano eliminati [1]. Questa rigidità nei pattern di movimento rendeva il gioco estremamente prevedibile, consentendo al giocatore di anticipare facilmente le mosse degli alieni dopo aver individuato lo schema. Inoltre, la totale assenza di reattività degli NPC faceva sì che le interazioni fossero esclusivamente meccaniche. Anche nei giochi di ruolo testuali o nelle prime avventure su computer, gli NPC erano estremamente rigidi. Spesso svolgevano un unico ruolo scriptato, come dare una quest o fungere da ostacolo, senza possibilità di deviazioni comportamentali. Per esempio, nel GDR primordiale **Akalabeth: World of Doom** (1979) di Richard Garriott, l'NPC Lord British era essenzialmente un dispensatore di missioni, un personaggio statico che impartiva al giocatore un compito, senza altre interazioni dinamiche [2]. La staticità del personaggio impediva qualsiasi tipo di reazione dinamica alle azioni del giocatore, riducendo l'immersione e la complessità delle interazioni. Gli anni '70 gli NPC non avevano intelligenza artificiale nel senso che intendiamo oggi: le loro azioni erano pre-programmate in anticipo e immutabili, fornendo sfide prevedibili e scriptate ai giocatori.

1980s: Miglioramenti nei movimenti e nelle linee di dialogo

Con gli anni '80, grazie a hardware più potente e a memorie più capienti, i comportamenti degli NPC diventarono gradualmente più vari sia nei movimenti sia nelle interazioni testuali. Nei giochi d'azione e arcade, i nemici iniziarono a muoversi in modi più sofisticati e meno prevedibili rispetto agli schemi fissi del decennio precedente. Un caso emblematico è **Pac-Man** (1980), i cui quattro fantasmi erano programmati con "personalità" differenti: ciascuno seguiva una logica di inseguimento diversa, rompendo la monotonia di nemici tutti uguali e rendendo l'IA di gioco più interessante [1]. Nonostante queste differenziazioni, i fantasmi di Pac-Man mantenevano comunque schemi deterministici. Una volta compreso il pattern, il giocatore poteva facilmente adottare una strategia vincente, e la reattività degli NPC restava limitata a risposte predefinite, senza adattarsi dinamicamente in tempo reale. Queste migliorie erano frutto di programmazione manuale ma offrirono agli NPC un minimo di variabilità e reattività maggiore di prima.

In contemporanea, nei giochi di avventura e di ruolo degli anni '80 si assistette a un grande arricchimento delle linee di dialogo e delle interazioni testuali con gli NPC. In precedenza, molti NPC si limitavano a ripetere una frase standard, ma a metà anni '80 divenne possibile inserire dialoghi più estesi e contestuali. Un titolo rivoluzionario fu **Ultima IV: Quest of the Avatar** (1985), che introdusse un innovativo sistema di dialogo a parole chiave. Il giocatore poteva digitare termini come "name" o "job" per interrogare un NPC, e ogni personaggio disponeva di molte risposte utili e pertinenti, ben oltre il singolo messaggio stereotipato dei giochi precedenti. Malgrado il sistema di dialogo a parole chiave abbia ampliato notevolmente le interazioni, rimaneva comunque vincolato a una serie finita di risposte predefinite che non si evolvevano dinamicamente in base alle continue interazioni del giocatore. Questo sistema diede vita a conversazioni più significative con gli NPC e permise di caratterizzarli meglio nel contesto del gioco. Verso la fine degli '80 si iniziarono anche a vedere NPC con routine giornaliere rudimentali: per esempio in **Ultima V: Warriors of Destiny** (1988) il mondo di gioco includeva un ciclo giorno/notte completo in cui gli abitanti seguivano un programma, i mercanti aprivano bottega al mattino, la sera andavano a dormire, e i dialoghi con questi personaggi riflettevano eventi e bisogni legati alla trama più complessa [3]. Queste routine giornaliere, sebbene aggiungessero un certo grado di variabilità, erano rigidamente programmate e non reagivano dinamicamente alle azioni del giocatore, mantenendo l'ambiente di gioco prevedibile e poco interattivo.

Sebbene elementari, questi avanzamenti tecnico-narrativi posero le basi per NPC più credibili, con movimenti meno meccanici e linee di dialogo più ricche.

1990s: Algoritmi goal-oriented

Negli anni '90 l'intelligenza artificiale nei videogiochi fece un salto di qualità, adottando approcci algoritmici più strutturati e goal-oriented per governare gli NPC. L'aumento della potenza di calcolo e l'avvento dei mondi 3D permisero di gestire routine più complesse: fu in questo periodo che tecniche classiche di AI come le macchine a stati finiti divennero uno standard per modellare il comportamento dei personaggi non giocanti [1]. In pratica, al posto di uno script lineare, all'NPC venivano definiti stati, ad esempio pattugliamento, allerta, inseguimento, fuga, e transizioni condizionate da eventi o stimoli, come "vede il giocatore" o "perde di vista il giocatore". Ciò consentiva reazioni più dinamiche: un nemico poteva cambiare obiettivo o strategia in base alla situazione, dando l'impressione di perseguire scopi propri. Questo approccio fu necessario anche per nuove tipologie di gioco emerse nei '90, come gli RTS, strategici in tempo reale, che mettevano a dura prova l'IA con molte unità da gestire, pathfinding su mappe complesse e decisioni economiche [1].

In termini pratici, gli NPC degli anni '90 risultarono più reattivi e adattabili. Nei FPS e nei giochi d'azione/avventura dell'epoca, i nemici iniziarono a mostrare comportamenti di squadra e tattiche elementari anziché attacchi suicidi predeterminati. Ad esempio, in **Half-Life** (1998) le guardie e i soldati controllati dall'IA cooperavano tra loro: lanciavano granate per stanare il giocatore, aggiravano ostacoli e cercavano di fiancheggiarlo, azioni tutte gestite tramite stati e controlli programmatici ma che simularono efficacemente un piano coordinato agli occhi del giocatore. Sebbene questa cooperazione apparisse avanzata, le reazioni degli NPC rimanevano basate su regole e script predefiniti. Una volta compresi i trigger del gioco, il giocatore poteva prevedere facilmente le strategie nemiche, riducendo così l'imprevedibilità delle interazioni. Allo stesso modo, i giochi d'avventura tridimensionali di fine decennio implementarono NPC con routine più contestuali. In **The Legend of Zelda: Ocarina of Time** (1998), primo Zelda in 3D, i villaggi erano popolati da personaggi con semplici obiettivi quotidiani: di giorno la piazza del mercato brulica di NPC che conversano e svolgono compiti, mentre di notte le stesse aree si svuotano o compaiono guardie e mostri differenti. I personaggi reagiscono anche allo stato del protagonista, creando l'illusione di una comunità viva con proprie agende. Sotto il cofano, si trattava ancora di script e trigger condizionali, ma l'effetto era che gli NPC sembravano perseguire obiettivi invece di seguire un copione fisso. Nonostante gli sforzi per creare una comunità viva, le reazioni degli NPC erano determinate da trigger specifici. Questo significava che le risposte alle azioni del giocatore erano spesso prevedibili e non permettevano una vera evoluzione dinamica della narrazione.

2000s: Statistical Learning e routine avanzate

Nel decennio 2000-2009, con l'ulteriore evoluzione dell'hardware e l'eredità della ricerca accademica, i giochi iniziarono a dotare gli NPC di routine ancor più sofisticate, talvolta descritte come basate su statistical learning o su simulazioni complesse del mondo di gioco. In pratica, anziché scriptare minuziosamente ogni comportamento, gli sviluppatori definirono regole generali e priorità che permettessero agli NPC di prendere decisioni autonomamente in base alla situazione, spesso introducendo elementi di casualità controllata per evitare ripetitività. Un esempio emblematico è il sistema **Radiant AI** sviluppato da Bethesda per **The Elder Scrolls IV: Oblivion** (2006). Con Radiant AI, a ciascun NPC vengono assegnati obiettivi di alto livello – ad esempio "alle 14:00 devi pranzare" – e una serie di possibili azioni per soddisfarli; sta poi all'NPC decidere *come* raggiungere lo scopo all'interno del mondo simulato [4]. Ciò significa che invece di programmare manualmente "vai alla taverna X e mangia il cibo Y alle 14:00", gli sviluppatori impostano solo il bisogno e le risorse disponibili, lasciando all'IA la scelta: un personaggio affamato potrebbe cercare una mela nel suo inventario, oppure andare a comprare del cibo, o persino rubarlo se è disperato, a seconda delle opportunità e della personalità assegnata. L'assenza di script individuali rigidi per ogni NPC rese possibile un mondo di gioco molto più ampio e variabile rispetto al passato. In Oblivion, infatti, gli NPC non rimangono sempre fermi negli stessi posti a ripetere le stesse azioni: grazie a questa IA radiante li si può vedere compiere routine quotidiane diversificate e reagire in modo più credibile agli stimoli (come conversazioni contestuali

tra NPC che il giocatore può ascoltare, opportunità di quest dinamiche, ecc.), il tutto generato sistemicamente anziché scriptato caso per caso. Nonostante questa maggiore variabilità, Radiant AI restava vincolato da regole preimpostate e probabilità controllate. In certe situazioni, ciò portava gli NPC a ripetere schemi comportamentali, rendendo prevedibili le loro reazioni alle azioni del giocatore. In generale, però, la maggior parte dei giochi anni 2000 continuò a basarsi su algoritmi deterministici avanzati, macchine a stati finite evolute, behavior trees, pianificazione GOAP introdotta in titoli come **F.E.A.R.** nel 2005, più che su vero machine learning. Ciò che cambiò significativamente fu la scala e la complessità delle simulazioni: mondi aperti enormi come quelli di **Grand Theft Auto** o **The Elder Scrolls** richiedevano sistemi che gestissero in modo semi-automatizzato decine o centinaia di NPC con le loro agende, e l'industria sviluppò soluzioni ad hoc sfruttando analisi statistiche, randomizzazione e scripting emergente.

2010s: Machine Learning e reti neurali

Negli anni 2010, la ricerca sull'intelligenza artificiale e il machine learning hanno iniziato a convergere più decisamente con lo sviluppo dei videogiochi, con l'obiettivo di creare NPC sempre più credibili e reattivi. La potenza di calcolo ormai elevatissima delle nuove console e PC, unita ai progressi nelle reti neurali e nel trattamento dei dati, aprì la porta a NPC dotati di qualche forma di adattamento automatico. In pratica, durante questo decennio i giochi di punta combinarono grandi quantità di scripting tradizionale con sistemi AI avanzati che conferivano ai personaggi una *memoria* e variabilità senza precedenti. Un esempio innovativo è il **Nemesis System** introdotto da Middle-earth: Shadow of Mordor (2014). Questo sistema generava proceduralmente nemici unici, i capitani Orchi, che **ricordavano** gli scontri avuti con il giocatore e ne tenevano conto nelle interazioni successive. Se un orco fosse sopravvissuto a un incontro, sarebbe potuto tornare più tardi ostentando cicatrici e vantandosi di aver affrontato il protagonista; se addirittura fosse riuscito a uccidere il giocatore, sarebbe avanzato di grado e sviluppava abilità aggiuntive. Il gioco creava una storia personale tra il giocatore e certi NPC nemici, in maniera non scriptata ma guidata da un sistema algoritmico [5]. Questo rendeva ogni partita diversa: due giocatori difficilmente sperimentavano lo stesso identico "nemico giurato", poiché gli Orchi nemici venivano plasmati dalle azioni del giocatore in tempo reale. Pur non utilizzando reti neurali, il Nemesis System è considerato un precursore nell'offrire NPC adattivi e con una parvenza di memoria e apprendimento all'interno di una struttura di gioco tradizionale.

Verso la fine degli anni 2010, titoli **open-world** ambiziosi portarono il realismo degli NPC a livelli altissimi, spesso servendosi anche di analisi dei dati e comportamenti emergenti che sfiorano il machine learning. **Red Dead Redemption 2** (2018) di Rockstar Games ne è un caso notevole: il mondo di RDR2 brulica di migliaia di NPC (umani e animali) che agiscono indipendentemente secondo regole e bisogni propri, dando vita a un ecosistema credibile e reattivo [6]. Ogni NPC umano ha una sorta di identità: compie lavori, reagisce ai cambiamenti ambientali e soprattutto "ricorda" le interazioni con il giocatore. Ad esempio, se Arthur maltratta o ferisce un personaggio, quel NPC conserverà rancore e, in un incontro successivo, potrà rivolgersi a lui in modo ostile o intimorito facendo riferimento all'evento passato. Viceversa, trattare bene un NPC può portarlo a fidarsi di più. Questo effetto di memoria persistente fa sì che i personaggi di RDR2 appaiano molto umani.

2.2 Futuro degli NPC: l'era degli LLM

L'avvento degli **LLM** sta aprendo orizzonti nuovi per la progettazione degli NPC. Un esempio pionieristico di impiego degli LLM nel gaming è **AI Dungeon** (2019), un gioco di avventura testuale interamente alimentato da IA. AI Dungeon utilizza un modello linguistico addestrato per generare in tempo reale descrizioni, dialoghi e azioni in risposta agli input del giocatore [7]. In pratica, funge sia da Dungeon Master che da NPC: il giocatore digita qualsiasi azione o frase, e l'IA improvvisa la prosecuzione della storia, interpretando il ruolo di tutti i personaggi non giocanti e del mondo stesso. Il risultato è un'esperienza con **possibilità virtualmente infinite**, dove la narrazione si adatta liberamente all'immaginazione del giocatore senza essere vincolata da script predefiniti. AI Dungeon è stato uno dei primi casi di **gioco completamente generativo**, dimostrando il potenziale degli LLM nel creare NPC testuali credibili e variazioni narrative imprevedibili. Se un giocatore in AI Dungeon decide di chiedere indicazioni a un locandiere o insultare un re, l'IA on the fly inventerà la risposta di quell'NPC in modo coerente col contesto e con un linguaggio naturale. Questo approccio evidenzia come gli LLM possano dare ai personaggi una parvenza di **creatività e comprensione linguistica** mai vista prima nei giochi.

Sulla scia di AI Dungeon, sono nati molti esperimenti e progetti che integrano modelli di linguaggio per dare voce e personalità dinamica agli NPC. Ad esempio, modder e sviluppatori indipendenti hanno iniziato a collegare **ChatGPT** a giochi esistenti. Un caso famoso è **Skyrim AI**: alcuni mod per The Elder Scrolls V: Skyrim hanno permesso di sostituire i dialoghi fissi degli NPC con risposte generate da ChatGPT in tempo reale [8]. In tali mod, il giocatore può effettivamente parlare con qualsiasi personaggio usando la voce o la tastiera, e l'NPC risponderà in linguaggio naturale, tenendo conto del contesto di gioco. È impressionante vedere un NPC di Skyrim che, invece di ripetere una linea registrata, può discutere liberamente col giocatore: chiedere aiuto, commentare il meteo o raccontare storie, il tutto generato dall'IA al momento.

Il termine "**Dungeon AI**" a cui si fa riferimento può essere inteso come questa nuova ondata di esperienze in cui l'IA generativa funge da motore creativo. Non solo AI Dungeon, ma anche altri prototipi stanno esplorando NPC generati al volo. Ad esempio, gli **Stanford Generative Agents** sono un esperimento accademico recente in cui decine di agenti NPC, ciascuno dotato di un LLM e di una memoria di lungo termine, simulano la vita quotidiana in un villaggio virtuale (una sorta di *The Sims* potenziato dall'IA). Questi agenti pianificano la propria giornata, interagiscono tra loro in linguaggio naturale e ricordano eventi passati, mostrando comportamenti credibilmente umani, il tutto senza script predeterminati, ma emergendo dall'architettura di IA generativa. È un campo di ricerca ancora agli inizi, ma promette NPC **completamente autonomi sia nell'azione che nel dialogo**.

NVIDIA ACE: NPC con intelligenza generativa

Anche le grandi aziende tech si stanno muovendo per portare gli LLM dentro i videogiochi. Una delle iniziative di spicco è **NVIDIA ACE (Avatar Cloud Engine)**, una suite di tecnologie annunciata nel 2023 pensata proprio per creare NPC avanzati con AI generativa. NVIDIA ACE è un servizio piattaforma che mette insieme vari componenti: modelli di linguaggio, modelli per la voce (riconoscimento e sintesi) e modelli per animazioni facciali, accelerati dalle GPU RTX, al fine di **dare vita a personaggi digitali credibili** [9]. L'idea è di fornire agli sviluppatori videogiochi una sorta di middleware AI chiavi in mano: invece di dover integrare manualmente ChatGPT, un TTS, ecc., ci si può appoggiare ad ACE che offre microservizi specializzati per gli NPC.

In pratica, ACE combina modelli linguistici per la conversazione con reti neurali per la visione e modelli decisionali rapidi in modo da permettere a un NPC di comportarsi quasi come un giocatore umano.

NVIDIA ha mostrato demo impressionanti: celebre è la "ramen shop demo" presentata al Computex 2023, dove il giocatore poteva dialogare liberamente con un NPC gestore di ramen in un'ambientazione cyberpunk. L'NPC, chiamato **Kai**, rispondeva con voce sintetizzata ma espressiva a domande anche inaspettate, ricordando il contesto della conversazione e mostrando animazioni facciali convincenti sincronizzate al parlato. Dietro le quinte, Kai era mosso da un LLM per comprendere e generare il testo, da **NVIDIA Riva** per la voce e da **Audio2Face** per animare in tempo reale le espressioni facciali.

Inworld AI: creare NPC con personalità via LLM

Un altro attore di spicco in questo campo è **Inworld AI**, una startup che offre una piattaforma per **sviluppare NPC avanzati alimentati dall'intelligenza artificiale**. Inworld fornisce un **Character Engine** basato su LLM: gli sviluppatori possono “configurare” un personaggio specificando una lore, un backstory, obiettivi, emozioni, nonché conoscenze e stile di dialogo, e poi lasciare che sia l'IA a generare in tempo reale il comportamento e la conversazione di quell'NPC. Inworld integra anche percezione sensoriale e memoria a lungo termine delle interazioni. L'ambizione è creare NPC che **sembrino davvero vivi e unici**, senza dover sceneggiare a mano ogni loro frase.

Un esempio concreto dell'utilizzo di Inworld è il progetto **Ubisoft NEO NPC** presentato nel 2024. Ubisoft Paris ha collaborato con Nvidia e Inworld per un prototipo di NPC di nuova generazione [10]. In questo prototipo, hanno realizzato due personaggi chiamati Bloom e Iron. Grazie alla tecnologia di Inworld, ciascuno di questi NPC ha un **background completo e uno stile conversazionale definito** dal team narrativo Ubisoft, e possiede anche una **conoscenza intrinseca dell'ambiente circostante**. Il risultato mostrato nei tre scenari dimostrativi di NEO NPC è stato impressionante: Bloom e Iron riuscivano a percepire il contesto, ricordare le conversazioni fatte con il giocatore, e adattare le proprie scelte di dialogo e animazione in modo coerente.

2.3 Ricerche Accademiche e Sviluppi sugli NPC basati su LLM

What if Red Can Talk? Dynamic Dialogue Generation Using Large Language Models

Il primo studio di Nananukul e Wongkamjan [11] affronta il problema della scarsa interazione dinamica nei giochi di ruolo (RPG), proponendo l'utilizzo dei modelli linguistici di grandi dimensioni come GPT-4 per generare automaticamente dialoghi filler per personaggi noti, come Cloud Strife in Final Fantasy VII Remake o Red in Pokémon, i quali normalmente restano muti o sono limitati a dialoghi pre-scriptati. L'approccio combina GPT-4 con knowledge graph costruiti partendo da informazioni strutturate estratte dai wiki dei giochi sotto forma di triplette RDF (soggetto-predicato-oggetto), che descrivono le relazioni e caratteristiche dei personaggi e del mondo di gioco. Questi dati vengono forniti a GPT-4 attraverso prompt specifici, consentendo così al modello di produrre risposte coerenti con la personalità dei personaggi e la situazione corrente. Nei test effettuati, GPT-4 ha dimostrato capacità significative nella generazione di dialoghi appropriati al contesto immediato, ma con una preferenza a enfatizzare elementi positivi della personalità, risultando a volte incoerente rispetto a personaggi originariamente cupi o freddi come Cloud. Nel caso specifico del protagonista Pokémon Red, sono state esplorate cinque personalità diverse, come allenatore maturo, novizio, loquace, timido, sicuro di sé, generando un totale di 120 dialoghi con NPC chiave come Brock o il Professor Oak. I risultati preliminari, basati su valutazioni umane, indicano che GPT-4 modula efficacemente lo stile linguistico in funzione della personalità esplicita del personaggio; in particolare, personalità marcate come "loquace" o "timida" hanno ricevuto valutazioni superiori, mentre quella più sottile e sfumata è risultata meno convincente. Tra i limiti identificati emergono dunque la difficoltà dei modelli a rappresentare fedelmente tratti di personalità meno marcati e l'incompleta integrazione dei dettagli enciclopedici dal knowledge graph nei dialoghi generati. Gli autori suggeriscono, pertanto, la possibilità di migliorare l'accuratezza e la fedeltà dei personaggi tramite un fine-tuning supervisionato mirato, da applicare in studi futuri con una più ampia varietà di giochi e valutazioni quantitative approfondite.

Leveraging Pre-trained Language Models for Interactive NPC Dialogue Generation

Il secondo studio, di Garcia e Lee [12], si concentra invece sull'utilizzo dei modelli di linguaggio pre-addestrati, come GPT-2 e GPT-3, per generare dialoghi dinamici degli NPC in situazioni interattive all'interno di videogiochi. L'obiettivo principale è superare i limiti dei dialoghi statici, offrendo risposte più credibili e contestualmente pertinenti alle azioni del giocatore e agli eventi del mondo virtuale. Gli autori esplorano due strategie principali: il fine-tuning dei modelli linguistici sui dati specifici del gioco, ad esempio, lore, quest, e dialoghi esistenti, e il prompting avanzato, che consiste nel fornire al modello informazioni dettagliate sul contesto attuale e sulla personalità dell'NPC per condizionare la generazione delle risposte. Un elemento centrale è l'attribuzione di profili caratteriali definiti agli NPC, consentendo al modello di variare coerentemente lo stile comunicativo a seconda del ruolo interpretato. I risultati sperimentali mostrano che, quando ben adattati al contesto di gioco, anche modelli linguistici di dimensioni moderate producono dialoghi di qualità paragonabile a quelli di modelli molto più grandi, sottolineando l'importanza di una conoscenza specifica piuttosto che della mera grandezza del modello. Valutazioni effettuate con utenti umani indicano che le frasi generate sono spesso indistinguibili da quelle scritte manualmente dagli sviluppatori, evidenziando coerenza narrativa e varietà lessicale superiori rispetto ai tradizionali script ripetitivi. Il paper riconosce limiti importanti, tra cui la gestione del contesto a lungo termine nelle conversazioni, problematica data la memoria limitata dei modelli Transformer, oltre al rischio di generazione di contenuti inesatti o fuori tema, che possono minacciare la coerenza del lore del gioco. Gli autori evidenziano inoltre la necessità di considerare limiti pratici, come la latenza di risposta e il consumo di risorse computazionali in contesti di gioco real-time. Pur confermando l'efficacia e le potenzialità dell'approccio proposto, il lavoro suggerisce ulteriori sviluppi tecnici, come l'utilizzo di sistemi di memoria esterna o knowledge base, per garantire la persistenza e la precisione delle conversazioni in-game e gestire meglio eventuali deviazioni stilistiche indesiderate.

Memory Repository for AI NPC

Quest'ultimo studio [13] presenta il MemoryRepository, un sistema di memoria a lungo termine per NPC basati su LLM, focalizzandosi sull'equipaggiare gli agenti virtuali con una memoria persistente e "umanizzata". Secondo gli autori, uno dei principali limiti degli LLM applicati ai personaggi nei giochi è la mancanza di memoria duratura: senza un metodo per immagazzinare informazioni, gli NPC non ricordano a lungo termine le interazioni precedenti, impedendo dialoghi estesi e coerenti. Per affrontare questa sfida, Zheng et al. suggeriscono una struttura mnemonica ispirata al processo cognitivo umano, distinguendo memoria a breve termine, ricca ma temporanea, e memoria a lungo termine, sintetica e durevole. Concretamente, il MemoryRepository conserva in forma integrale le conversazioni recenti (breve termine) sintetizzando e trasferendo invece le informazioni apprese nella memoria di lungo periodo dell'NPC, permettendo così ai personaggi di "dimenticare" dettagli superflui mantenendo però il senso complessivo degli eventi passati. Questo approccio di oblio selettivo e consolidamento migliora la naturalezza delle interazioni: l'NPC può ricordare eventi significativi o rapporti costruiti precedentemente senza necessariamente conservare ogni singolo dettaglio, proprio come farebbe una persona reale. Tale meccanismo è stato implementato in maniera indipendente dal modello linguistico sottostante, funzionando efficacemente sia con LLM proprietari che open-source, ed è stato applicato ad alcuni personaggi esemplificativi, con stili dialogici adattati ai loro ruoli narrativi. I test sperimentali indicano che un NPC con MemoryRepository sostiene dialoghi più lunghi e coerenti, adattandosi progressivamente alle interazioni col giocatore: durante gli esperimenti, questi NPC hanno dimostrato la capacità attiva di sintetizzare e scartare conversazioni passate, conservando una continuità logica e mostrando comportamenti più realistici nel lungo termine. Questa proposta costituisce un significativo progresso, integrando una memoria artificiale negli NPC basati su modelli di linguaggio e migliorandone notevolmente la capacità di interagire nel tempo e la credibilità delle loro risposte nel gioco.

3. IDEAZIONE DEL SISTEMA

3.1. Concetti di base

3.1.1. Spiegazione e origine dell'idea

L'idea alla base del progetto prende ispirazione dalle tecniche narrative impiegate dai Game Master nei giochi di ruolo, figure essenziali nella creazione e nella gestione di mondi complessi popolati da numerosi personaggi interconnessi. Nei giochi di ruolo, i Game Master devono monitorare e aggiornare costantemente le relazioni tra i personaggi per mantenere la narrazione coerente e dinamica al seguito di incontri o eventi avvenuti tra i personaggi stessi. Per fare ciò, essi spesso fanno utilizzo di **mappe concettuali**, strumenti grafici che rappresentano visivamente le interazioni e le relazioni fra i protagonisti del gioco.

Le mappe concettuali non sono semplicemente dei grafi statici; essi sono sistemi dinamici e flessibili, in cui ogni **nodo** rappresenta un'entità, che può essere un singolo personaggio ma anche estendersi a rappresentare un intero gruppo, e ogni **link** simboleggia il legame o la relazione che intercorre tra tali entità. Questi link non sono fissi, bensì possiedono attributi come tipo e intensità, che possono variare in tempo reale in funzione degli eventi narrativi e delle azioni svolte dai personaggi.

La Dinamica delle Mappe Concettuali

Si immagini, ad esempio, una mappa concettuale composta da tre entità: Alice, Bob e Carla.

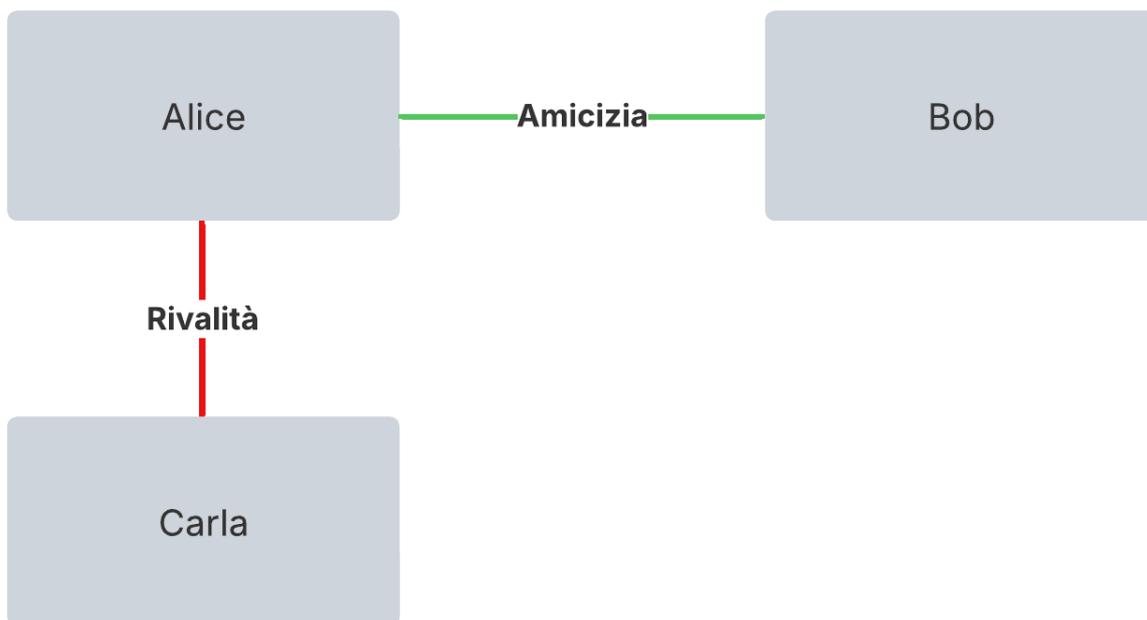


Figura 1: Mappa che descrive un semplice sistema di relazioni

In una fase iniziale, il link tra Alice e Bob potrebbe rappresentare una forte relazione di collaborazione o amicizia, mentre il link tra Alice e Carla potrebbe indicare una relazione di rivalità o conflitto. La natura di questi collegamenti potrebbe però cambiare durante il progredire della narrazione, se Alice compie un'azione che tradisce la fiducia di Bob, oppure, al contrario, rafforza il legame di lealtà, il peso del link che li unisce si modificherà in maniera proporzionale. Analogamente, un'azione aggressiva o un atto di riconciliazione tra Alice e Carla potrà intensificare o attenuare il grado di conflitto rappresentato dal loro collegamento.

Questo meccanismo dinamico permette alla mappa concettuale di aggiornarsi in tempo reale, riflettendo le variazioni del contesto narrativo e delle interazioni tra i personaggi. Ogni azione eseguita da un NPC diventa un **evento** in grado di modificare la struttura relazionale dell'intero sistema. Tale approccio non solo garantisce una maggiore fluidità nella gestione delle dinamiche narrative, ma consente anche di prevedere e simulare le ripercussioni a catena che possono scaturire da un singolo intervento all'interno del sistema.

Astrazione in Gruppi e Propagazione Inter-Gruppo

Un aspetto importante del modello è la possibilità di **astrazione in gruppi**. Invece di gestire ogni nodo della mappa concettuale, in questo caso un NPC, come un'entità isolata, è possibile organizzare l'intero sistema in gruppi che possono rappresentare insiemi di personaggi con caratteristiche, obiettivi o ruoli comuni. Questi gruppi, però, non si limitano a piccole aggregazioni di NPC, ma possono essere estesi ad entità più complesse come fazioni, nazioni, città o altre strutture sociali. Ad esempio, in un'ambientazione narrativa potremmo distinguere fra un gruppo di **Eroi** e un gruppo di **Oscuri Signori**. Tali gruppi, oltre a raccogliere al loro interno NPC con tratti simili, sono anche collegati fra loro da relazioni che possono essere di alleanza, rivalità o persino conflitto aperto.

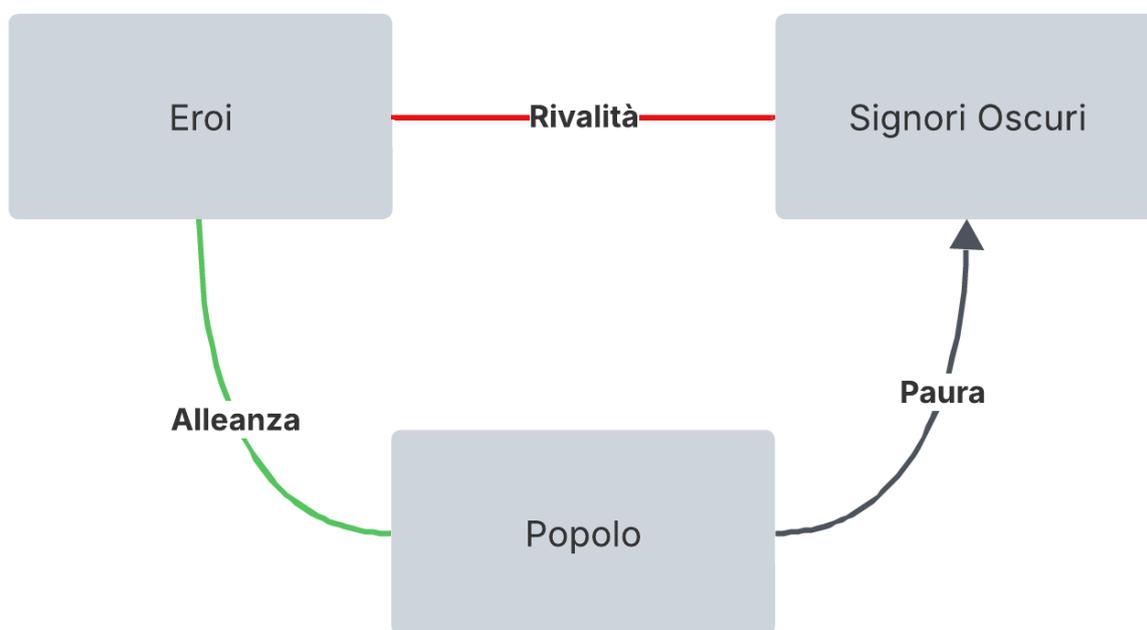


Figura 2: Mappa concettuale

Questa astrazione permette di gestire le dinamiche in modo più scalabile e modulare: invece di gestire singolarmente ogni relazione, il sistema può considerare le interazioni a livello di gruppo, facilitando la propagazione degli eventi.

Ad esempio, immaginiamo che all'interno del gruppo degli Eroi il giocatore compia un'azione di grande coraggio. Questo gesto rafforza il legame interno tra il gruppo degli Eroi e il giocatore, aumentando la fiducia e la cooperazione del gruppo rispetto al giocatore. Tuttavia, l'evento non si limita a influenzare il rapporto diretto: esso ha anche ripercussioni sui gruppi direttamente collegati agli Eroi. Le fazioni o le entità in stretta relazione con il gruppo degli Eroi iniziano a rivedere la loro percezione del giocatore, adattando il loro atteggiamento in base all'azione coraggiosa appena compiuta. L'azione del giocatore genera una reazione a catena che non solo consolida il rapporto con gli Eroi, ma influenza anche il modo in cui gli altri gruppi, collegati o alleati agli Eroi, interagiscono e collaborano con lui.

Il Ruolo degli Eventi nella Propagazione delle Informazioni

In questo modello, il concetto di **propagazione degli eventi** assume un ruolo centrale. La generazione di eventi all'interno del sistema è il meccanismo attraverso il quale le azioni individuali vengono trasformate in dei pacchetti capaci di modificare la rete relazionale e, propagandosi all'interno dei gruppi, permettere agli NPC di venire a conoscenza di eventi "indiretti", ovvero di eventi avvenuti tra altri membri del sistema. Ogni evento porta con sé informazioni specifiche, come la natura dell'azione (ad esempio, cooperazione, aggressione, tradimento), la sua intensità e le persone coinvolte. Queste informazioni vengono elaborate dal sistema per aggiornare dinamicamente il peso e la tipologia dei link nella mappa concettuale.

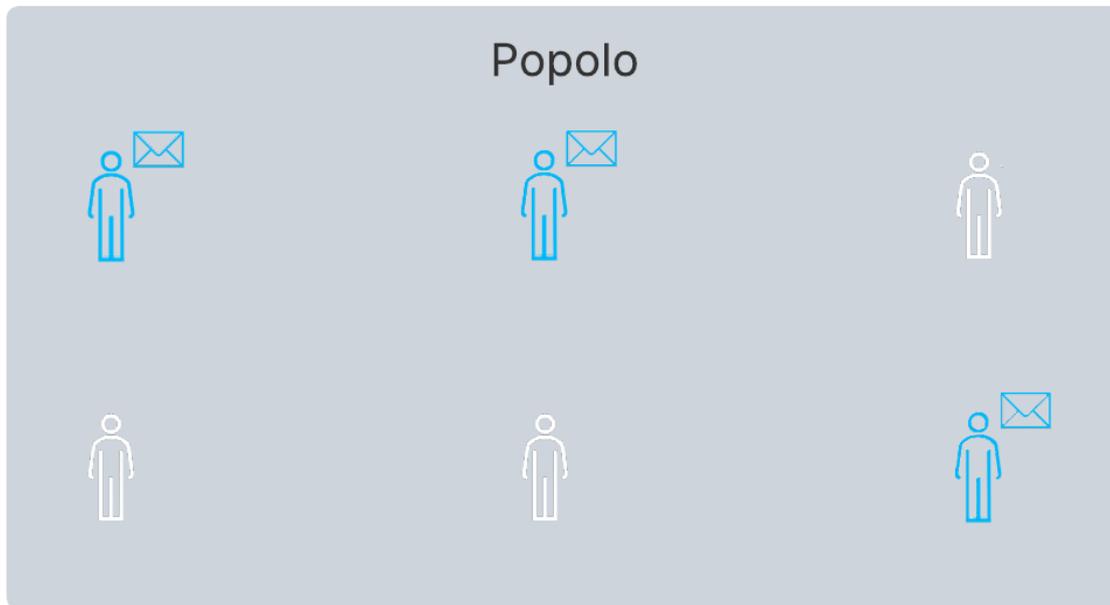


Figura 3: Esempio di propagazione all'interno del gruppo

Concetto di Pensiero sul Giocatore e sui Gruppi

Un concetto che emerge da questo modello è quello del **pensiero sul giocatore**. Ogni NPC non solo reagisce in base alla propria personalità e al contesto in cui si trova, ma sviluppa anche un proprio "giudizio" o percezione del giocatore. Questo pensiero, che si accumula e si modifica nel tempo in base alle esperienze personali e agli eventi indiretti, influenza direttamente il comportamento dell'NPC durante le interazioni con il giocatore. Ad esempio, un NPC che ha un pensiero negativo a causa di azioni ostili del giocatore potrebbe adottare un atteggiamento difensivo o addirittura ostile nelle fasi successive, mentre un NPC che ha percepito comportamenti positivi potrebbe mostrare atteggiamenti di amicizia e fiducia nei suoi confronti.

Parallelamente, gli NPC sviluppano anche un **pensiero sugli altri gruppi**. Questo aspetto riguarda la capacità degli NPC di formarsi opinioni riguardo a interi insiemi o comunità e di modulare il proprio comportamento in base a tali opinioni. Il modo in cui un NPC interagisce con una persona che non conosce e appartenente a un determinato gruppo può variare notevolmente a seconda del pensiero che ha maturato nei confronti di quel gruppo. Se, ad esempio, un NPC ha un pensiero particolarmente negativa del gruppo degli Oscuri Signori a causa di esperienze passate o eventi ricevuti, potrebbe essere più incline a esprimere diffidenza, a utilizzare un linguaggio aggressivo o a prendere decisioni che penalizzino chiunque provenga da quel gruppo. Al contrario, un pensiero positivo potrebbe portare a interazioni più pacifiche e collaborative.

Questa doppia dimensione del "pensiero", sia sul giocatore che sugli altri gruppi, rappresenta un elemento fondamentale del sistema. Infatti, mentre le mappe concettuali e la propagazione degli eventi offrono una struttura per comprendere e gestire le relazioni e le dinamiche interne al sistema, il pensiero sul giocatore e sugli altri gruppi introduce un livello di reattività agli eventi che rende ogni interazione unica e contestuale.

Ogni NPC, infatti, diventa non solo un semplice bot che risponde alle domande, ma un agente dotato di una “memoria” e di giudizio, che si adatta in maniera dinamica al mondo in continuo movimento in cui vive.

I Pilastri del Sistema e i Benefici del Modello

Il modello descritto si fonda su tre pilastri fondamentali: la **mappa concettuale**, la **generazione e propagazione degli eventi**, e il **meccanismo dei pensieri sul giocatore e sui gruppi**. Questi elementi interconnessi costituiscono il cuore del sistema, offrendo una struttura dinamica capace di gestire le interazioni in tempo reale in ambienti narrativi o simulativi.

Il primo pilastro, la **mappa concettuale**, rappresenta lo scheletro strutturale del sistema. Essa permette di visualizzare in maniera immediata e intuitiva le relazioni tra i vari nodi, che possono rappresentare singoli NPC o aggregazioni più complesse quali fazioni, nazioni o città. La dinamicità della mappa consente di adattarsi in modo flessibile agli aggiornamenti in tempo reale, garantendo che il sistema rispetti fedelmente le evoluzioni e le interazioni sociali all'interno dell'ambiente.

Il secondo pilastro riguarda la **generazione e propagazione degli eventi**. Ogni azione compiuta da un NPC viene interpretata come un evento che si diffonde all'interno della rete relazionale. Gli eventi fungono da vettori di influenza, capaci di influenzare la natura dei link tra i nodi, aggiornando la mappa concettuale, il comportamento dei singoli personaggi nei confronti del giocatore o di altri. In questo modo, un singolo intervento può avere ripercussioni estese, contribuendo a creare un ambiente reattivo e in continua evoluzione.

Il terzo pilastro è rappresentato dal **meccanismo dei pensieri sul giocatore e sui gruppi**. Ogni NPC sviluppa una propria “opinione” in base alle interazioni passate e agli eventi indiretti da lui ricevuti, che si traduce in un comportamento specifico nei confronti del giocatore o di altri personaggi provenienti da altri gruppi. Tale capacità di “pensare” in maniera autonoma introduce una componente di personalizzazione e profondità che arricchisce l'esperienza narrativa, rendendo ogni interazione unica e contestuale.

I benefici derivanti da questo modello sono molteplici. In primis, la capacità di visualizzare e aggiornare dinamicamente le relazioni tramite la mappa concettuale permette di monitorare e ottimizzare le interazioni in tempo reale, facilitando la gestione di sistemi complessi. In più, la propagazione degli eventi garantisce che le azioni di un singolo NPC possano influenzare l'intero ecosistema, creando dinamiche di reazione a catena rendono l'ambiente più vivo e immersivo.

3.1.2. Introduzione delle fondamenta: eventi, mappa concettuale e pensiero sul giocatore e sul gruppo

In questo capitolo si analizzano nel dettaglio ciascuno dei pilastri introdotti precedentemente, chiarendo come essi siano strutturati e quali sono le informazioni specifiche che li caratterizzano. Verranno approfonditi gli **eventi**, la **mappa concettuale** e il **pensiero**, sia sul giocatore che sugli altri gruppi.

Gli Eventi

Approfondendo la composizione degli eventi abbiamo che essi sono così strutturati:

- **Tipologia:** Parametro che classifica l'evento in base alla natura dell'interazione, ad esempio se si tratta amicizia, di tradimento o di conflitto.
- **Importanza:** Un valore che quantifica il peso dell'evento. Un evento di grande importanza potrà influenzare in modo marcato lo stato emotivo dell'NPC, portando a cambiamenti comportamentali significativi.
- **Descrizione:** Una delimitazione testuale del contesto e la dinamica dell'interazione. Questa serve a contestualizzare l'evento e a renderlo interpretabile dall'NPC.
- **Persone Coinvolte:** Una lista di persone che hanno partecipato alla conversazione o all'interazione. Questo elemento è cruciale per comprendere la diffusione dell'evento nella rete relazionale.
- **Impatto sull'NPC:** Un indicatore che misura la variazione nello stato d'animo o nel comportamento dell'NPC in seguito all'evento. Tale impatto è determinante per aggiornare i valori interni dell'agente e per influenzare le sue future interazioni.

Attraverso questi elementi, ogni evento diventa un pacchetto informativo in grado di innescare una serie di reazioni a catena, contribuendo a modificare dinamicamente la struttura delle relazioni all'interno della mappa concettuale.

La Mappa Concettuale

Come introdotto precedentemente, la mappa concettuale consente di visualizzare dinamicamente le relazioni. In particolare, la mappa è composta da:

- **Nodi:** Ogni nodo rappresenta un gruppo aggregato di NPC.
- **Link:** I collegamenti tra i nodi sono definiti da attributi che indicano il tipo e l'intensità della relazione. Questi valori si aggiornano in tempo reale in risposta agli eventi e alle interazioni che avvengono nel sistema.
- **Valori Proprietari dei Gruppi:** Per ogni gruppo, si definiscono due parametri fondamentali:
 - **Dimensione:** Il numero indicativo per la grandezza del gruppo, indicatore della sua rilevanza e della potenziale influenza.
 - **Affiatamento:** Un valore che esprime il grado di coesione e sintonia tra i membri del gruppo, riflettendo quanto efficacemente il gruppo possa reagire agli eventi interni ed esterni.

Questa struttura permette di avere una visione chiara e immediata dell'intero ecosistema, evidenziando come ogni variazione in un nodo o in un link possa influire sull'equilibrio complessivo delle relazioni.

Il Pensiero sul Giocatore

Il meccanismo del pensiero sul giocatore è una componente chiave fondamentale ed è composta da tre aspetti fondamentali:

- **Rispetto:** Indica il livello di ammirazione o riconoscimento che l’NPC attribuisce al giocatore. Ad esempio, se il valore è molto alto, l’NPC esprime una sincera ammirazione, condividendo informazioni personali in modo caloroso e dettagliato; al contrario, se è molto basso, il giocatore viene percepito con evidente disprezzo, tanto da indurre l’NPC a usare un tono diretto e irrispettoso evitando qualsiasi approfondimento personale.
- **Fiducia:** Misura quanto l’NPC si senta sicuro nell’interagire con il giocatore. Un valore elevato significa che il giocatore venga considerato affidabile, facilitando collaborazioni e scambi informativi. Invece, se il valore è significativamente basso, l’NPC diventa estremamente sospettoso, condividendo al massimo informazioni essenziali e cambiando rapidamente argomento per proteggersi da possibili tradimenti.
- **Notorietà:** Questo parametro quantifica la fama del giocatore. Valori positivi elevati portano l’NPC a interagire con entusiasmo e apertura, mentre una notorietà marcata da valori negativi induce l’NPC a mantenere un atteggiamento cauto, riservato e spesso critico nei confronti del giocatore.

Questi valori vengono aggiornati costantemente in seguito agli eventi e alle interazioni, e incidono direttamente sul comportamento dell’NPC.

Il Pensiero Sugli Altri Gruppi

Analogamente al pensiero sul giocatore, il pensiero sugli altri gruppi è composto da cinque valori diversi:

- **Cooperazione:** L’NPC valuta la disponibilità a collaborare con i membri di un gruppo. Se la cooperazione è elevata, l’NPC si mostra pronto per raggiungere obiettivi comuni; se, invece, la cooperazione risulta bassa, l’NPC tende a isolarsi o a sabotare iniziative, percependo tali proposte come poco vantaggiose.
- **Fiducia:** In questa dimensione l’NPC esprime il grado di affidabilità che attribuisce al gruppo. Una fiducia elevata fa emergere la convinzione che il gruppo sia onesto e collaborativo. Al contrario, una fiducia bassa induce l’NPC a mantenere una posizione di cautela, valutando ogni proposta con sospetto.
- **Paura:** La percezione del pericolo è determinata dal livello di timore nei confronti del gruppo. Se l’NPC prova una forte paura, il gruppo viene considerato una minaccia concreta, attivando comportamenti difensivi; se la paura è bassa, invece, l’NPC tende a vedere il gruppo come meno pericoloso o addirittura inoffensivo, rischiando così di sottovalutare eventuali rischi.
- **Stima:** Questo aspetto riflette il livello di rispetto e ammirazione attribuito al gruppo. Una stima elevata porta l’NPC a sostenere attivamente le iniziative e i progetti del gruppo, promuovendone la reputazione. Al contrario, una stima bassa induce l’NPC a minimizzare o ignorare le qualità positive del gruppo, manifestando un atteggiamento critico e distaccato.
- **Ostilità:** Infine, l’ostilità indica la predisposizione a comportamenti antagonisti. Un livello elevato di ostilità può portare l’NPC a reagire in modo conflittuale o a opporsi apertamente al gruppo, mentre una bassa ostilità favorisce un clima di convivenza pacifica e la risoluzione di eventuali divergenze tramite negoziazioni.

L’approfondimento di questi meccanismi di pensiero è fondamentale per garantire un comportamento degli NPC che risulti non solo reattivo ma anche coerente e contestualizzato. Infatti, il continuo aggiornamento dei valori relativi al pensiero sul giocatore e sugli altri gruppi consente di modellare un “background emotivo” dinamico, in cui ogni interazione contribuisce a plasmare la personalità virtuale dell’NPC. Questa caratteristica rende il sistema particolarmente adatto a simulare scenari narrativi complessi, dove la varietà delle risposte e la capacità di adattarsi alle nuove informazioni sono essenziali.

3.2 Generazione di un nuovo evento

La scelta progettuale di generare un evento al termine della conversazione tra il giocatore e l'NPC è stata adottata per semplificare il sistema complessivo e rendere più gestibile la propagazione delle informazioni. Con questa scelta, ogni conversazione conclusa produce un evento strutturato che raccoglie e sintetizza il contenuto della comunicazione, facilitando successivamente l'analisi, la classificazione e la diffusione dell'informazione nel sistema.

Descrizione dell'Evento

La descrizione viene generata riassumendo la conversazione dal punto di vista dell'NPC e rappresenta una sintesi narrativa che cattura i punti salienti del dialogo.

Per realizzare questa descrizione, si utilizza una richiesta a un modello LLM. Questo riceve come input l'intera conversazione, insieme al contesto in cui essa si è svolta e produce un riassunto che evidenzia gli elementi chiave e le emozioni emergenti durante lo scambio. In questo modo, la descrizione non è solamente una copia del dialogo, ma è una reinterpretazione soggettiva dell'NPC.

Tipologia e Importanza

La determinazione della tipologia e dell'importanza dell'evento è un aspetto critico della generazione degli eventi, in quanto questi elementi guideranno le fasi successive di propagazione e influenzeranno il modo in cui gli NPC gestiscono l'evento. Per questo scopo esistono due approcci principali:

1. Utilizzo di un classificatore:

In questo approccio, si impiega un modello di classificazione appositamente addestrato, al quale viene fornita in input l'intera conversazione tra il giocatore e l'NPC. Il classificatore analizza il contenuto testuale e semantico, estraendo caratteristiche chiave che permettono di assegnare in modo automatico una categoria per la tipologia dell'evento e di determinare il relativo livello di importanza.

2. Utilizzo di Tecniche NLP basate su Sentix:

In alternativa, il processo di determinazione può essere realizzato mediante tecniche di elaborazione del linguaggio naturale (NLP) che sfruttano dizionari di polarità e intensità. In questo contesto, Sentix rappresenta uno strumento fondamentale: si tratta di un dizionario che associa ad ogni parola un valore di polarità, indicante se la parola ha una connotazione positiva o negativa, e un valore di intensità, che ne quantifica la forza espressiva.

Calcolo della Polarità:

La polarità dell'evento viene calcolata effettuando una media delle polarità delle parole all'interno di una frase, e, successivamente, facendo la media dei risultati di tutte le frasi. Formalmente, se definiamo p_{ij} come la polarità della i -esima parola della j -esima frase e M_j il numero di parole nella j -esima frase, e N il numero totale delle frasi, la polarità complessiva P si viene definita come:

$$P = \frac{1}{N} \sum_{j=1}^N \left(\frac{1}{M_j} \sum_{i=1}^{M_j} p_{ij} \right) [-1, 1]$$

Il valore ottenuto indica se l'evento è positivo, negativo o neutrale e, in base a una keyword map, viene associata una tipologia specifica.

Calcolo dell'Importanza:

Analogamente, per l'importanza, se definiamo i_{ij} come l'intensità della i -esima parola della j -esima frase, l'importanza complessiva I viene definita mediante:

$$I = \frac{1}{N} \sum_{j=1}^N \left(\frac{1}{M_j} \sum_{i=1}^{M_j} p_{ij} \right)$$

Il valore medio risultante, compreso tra 0 e 1, viene poi utilizzato per categorizzare l'evento in una delle importanze disponibili.

Sebbene l'approccio basato su Sentix offra una modalità di analisi che considera la polarità e l'intensità di ogni parola, tale metodo tratta ogni termine in maniera indipendente, senza tenere conto del contesto più ampio della conversazione. Di conseguenza, può succedere che la somma dei contributi di singole parole non rifletta adeguatamente il tono o il significato reale della conversazione.

Il classificatore, invece, elabora l'intero contesto, considerando non solo il significato di singole parole, ma tiene conto anche del contesto e del modo in cui vengono dette. Questo approccio permette di avere una visione più accurata, che porta a una scelta della tipologia e dell'importanza dell'evento più vicina alla realtà.

Pertanto, pur riconoscendo i vantaggi di Sentix per l'analisi fine dei dettagli lessicali, si preferisce l'uso del classificatore per la generazione di tipologia e importanza, in quanto esso offre un controllo più contestuale, riducendo il rischio di errori derivanti dall'interpretazione isolata delle informazioni.

Generazione delle Statistiche

La generazione delle statistiche di un evento avviene analizzando il contesto della conversazione e il contenuto testuale scambiato durante l'interazione stessa. Anche in questo caso, come in precedenza, sono stati individuati due approcci principali:

1. Approccio basato su IA:

In questo metodo, il testo della conversazione, insieme alle informazioni sul contesto e alla storia pregressa tra NPC e giocatore, viene fornito in input a un modello LLM. Questo elabora i dati e restituisce un set di statistiche. L'uso di un modello IA consente di sfruttare tecniche avanzate di elaborazione del linguaggio naturale per catturare le sfumature della comunicazione. Ad esempio, il modello può considerare l'intonazione, il contesto conversazionale e persino elementi impliciti che potrebbero non essere immediatamente evidenti ad un'analisi superficiale. In questo modo, le statistiche generate riflettono in maniera dinamica e contestuale il contenuto della conversazione, garantendo un output che si adatta alle specificità di ogni interazione.

2. Approccio deterministico:

Le statistiche dell'evento possono essere assegnate in modo deterministico, basandosi su una serie di regole fisse che considerano l'importanza e la tipologia dell'evento. Ad esempio, si utilizza una struttura switch-case che valuta il tipo di evento e la sua importanza per determinare i valori delle statistiche. Questo approccio offre un controllo fine sui parametri, evitando che i valori generati escano da un range accettabile e garantendo una maggiore coerenza nel modello complessivo.

Dopo un'attenta analisi, si è optato per l'approccio deterministico. La scelta è stata motivata dal maggiore controllo che ci garantisce sulle statistiche, che evita l'assegnazione di valori eccessivi e potenzialmente distorsivi. Pur risultando più rigido e meno soggetto agli scambi tra NPC e giocatore, questo approccio assicura la coerenza e la stabilità del modello complessivo, elementi fondamentali per mantenere l'integrità del sistema.

3.3. Diffusione degli eventi: prima idea

Qui si analizzerà nel dettaglio come un evento, una volta generato, si diffonda all'interno del sistema. L'obiettivo è quello di definire un modello matematico in grado di rappresentare la complessità delle interazioni sociali in una serie di formule che, restituendo una probabilità, determinano se e come l'evento viene trasmesso da un NPC all'altro. Si introduce anche il concetto di velocità di diffusione, che traduce il ritmo al quale l'evento si propaga, integrando anche una componente temporale che consente di rendere il modello realistico, dato che, nella realtà, gli eventi non vengono trasmessi istantaneamente, ma richiedono tempo per propagarsi.

Approccio Deterministico vs. Probabilistico

In questo sistema la scelta tra un approccio deterministico e uno probabilistico è un aspetto cruciale per rappresentare la natura complessa delle interazioni sociali. In un modello deterministico, ogni variabile viene combinata in maniera fissa e, dato un insieme di input, l'output sarebbe sempre identico. Questo approccio presenta il vantaggio di una prevedibilità assoluta: se le condizioni sono note, l'output risulta esattamente calcolabile e ripetibile. Tale rigidità non cattura a pieno la complessità e l'imprevedibilità delle dinamiche sociali, dove piccole variazioni e fattori casuali possono influire notevolmente sulla propagazione di un evento.

Per modellare in maniera più realistica tali dinamiche, questo sistema adotta un approccio probabilistico. In questo quadro, si utilizzano delle formule che restituiscono una probabilità compresa tra 0 e 1, che indica la possibilità che un evento venga trasmesso da un agente all'altro. In questo modo si simula l'incertezza intrinseca delle interazioni sociali.

Propagazione degli Eventi all'Interno del Gruppo

La propagazione degli eventi all'interno di un gruppo è influenzata da una serie di fattori che rappresentano sia le caratteristiche del gruppo stesso sia la natura dell'evento. I principali parametri presi in considerazione sono:

- **Grandezza del gruppo (G):** Un valore che esprime quanto è grande un gruppo. Un gruppo di dimensioni notevoli offre maggiori opportunità di interazione e, quindi, una maggiore probabilità di diffusione, poiché ci sono più canali attraverso i quali l'evento può essere trasmesso.
- **Affiatamento (A):** Questo parametro misura il livello di coesione e sintonia tra i membri del gruppo. In gruppi con alto affiatamento, gli NPC sono più predisposti a reagire in maniera simile e a condividere le stesse informazioni emotive, facilitando così la propagazione dell'evento.
- **Importanza dell'evento (I):** Rappresenta il peso emotivo o narrativo dell'evento. Eventi di grande importanza tendono a generare reazioni più forti e a propagarsi più facilmente, in quanto influenzano maggiormente lo stato d'animo degli NPC.
- **Tipologia dell'evento (T):** Indica la natura dell'evento, che può variare da atti di cooperazione a gesti di tradimento, da manifestazioni di lealtà a segnali di conflitto. La tipologia fornisce un ulteriore livello di dettaglio che modula l'efficacia della diffusione.

Questi quattro parametri, tutti normalizzati in un intervallo [0, 1], vengono combinati per determinare la probabilità che l'evento venga trasmesso da un NPC all'altro all'interno dello stesso gruppo. La formula adottata è:

$$P_{interno} = 1 - e^{-G \cdot A \cdot I \cdot T}$$

La scelta di utilizzare la funzione esponenziale negativa, $1 - e^{-x}$, permette di garantire che la probabilità di diffusione non si annulli completamente anche se uno dei parametri è vicino a zero, consentendo un'attenuazione graduale, mantenendo la possibilità di trasmissione anche in condizioni meno favorevoli. In questo modo, ogni NPC valuta se ricevere l'evento confrontando un numero casuale (estratto uniformemente tra 0 e 1) con il valore $P_{interno}$, se il numero casuale è inferiore, l'evento viene trasmesso.

Propagazione degli Eventi all'Esterno del Gruppo

La propagazione degli eventi tra gruppi si basa su un meccanismo leggermente diverso in quanto, oltre a dover considerare i fattori interni già descritti, è necessario integrare ulteriori parametri che riflettano le caratteristiche delle relazioni tra gruppi differenti. Prima di tutto, è fondamentale stabilire una soglia: un evento viene considerato idoneo a essere trasmesso all'esterno solo quando è stato diffuso all'interno del gruppo fino a raggiungere quella soglia, che potrebbe essere, ad esempio, almeno la metà degli NPC presenti. Tale soglia assicura che l'evento abbia acquisito una rilevanza sufficiente e che la sua diffusione interna sia consolidata.

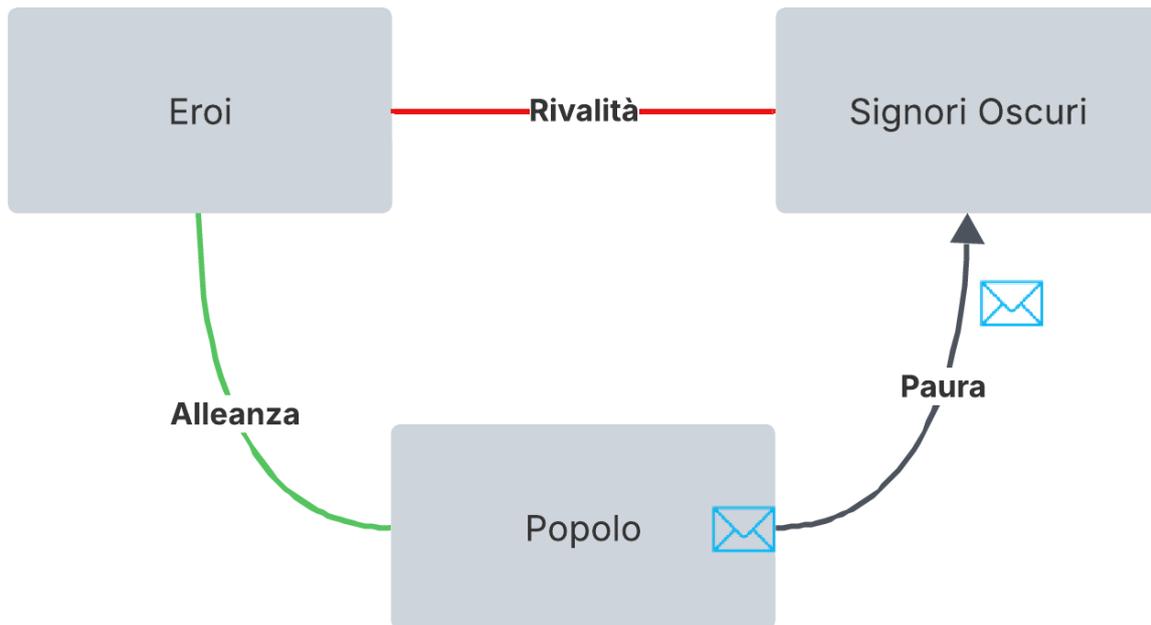


Figura 4: esempio di evento generato dentro il gruppo del popolo e poi propagato

Una volta superata questa soglia, la probabilità che l'evento venga trasmesso ad un altro gruppo viene determinata da una formula che integra i seguenti parametri:

- **Tipologia del link (L):** Indica la qualità del collegamento tra i gruppi. Valori elevati di L indicano un collegamento forte e consolidato, mentre valori bassi indicano relazioni deboli. La qualità del link influenza notevolmente la capacità dell'evento di essere trasmesso.
- **Distanza (D):** Misura la "distanza" tra i gruppi. Una distanza elevata comporta maggiori difficoltà di trasmissione, riducendo la probabilità che l'evento attraversi il confine.
- **Importanza dell'evento (I):** Come nel caso interno, questo parametro indica quanto l'evento sia rilevante dal punto di vista emotivo o narrativo, contribuendo a rafforzare la sua capacità di propagarsi.
- **Tipologia dell'evento (T):** Modula la probabilità in base alla natura qualitativa dell'evento.

La formula utilizzata per la diffusione esterna è:

$$P_{esterno} = 1 - e^{-L \cdot D \cdot I \cdot T}$$

Questa espressione mantiene una continuità con il modello interno e garantisce che la probabilità di trasmissione non scenda bruscamente a zero anche in presenza di condizioni non ottimali. La combinazione

moltiplicativa dei parametri L, D, I e T permette di modulare in maniera sensibile l'efficacia del collegamento tra gruppi. Ad esempio, un link di alta qualità (L elevato) e una distanza relazionale bassa (D ridotto) possono compensare una moderata importanza dell'evento, mantenendo la probabilità di diffusione elevata. Allo stesso tempo, se uno dei fattori risulta particolarmente sfavorevole, la funzione esponenziale agirà in maniera da ridurre progressivamente la probabilità, evitando il passaggio dell'evento.

Un altro aspetto usato in questo ambito è l'applicazione del concetto di **filter** e **weight** sui link. Quando un evento viene trasmesso tramite un link, il meccanismo di filter e weight interviene per adattare la ricezione dell'evento nel gruppo di destinazione in base alla relazione esistente tra i due gruppi.

Il **filter** ha il compito di scartare quegli eventi che, in base alla tipologia del link e dell'evento, non raggiungono una certa soglia di rilevanza. Ad esempio, se il link tra due gruppi è di bassa qualità o se l'evento fosse di bassa importanza, il filtro potrebbe determinare che l'evento debba essere ignorato, impedendo così una propagazione inutile o indesiderata. Il **weight**, invece, modifica le statistiche dell'evento in base alla qualità del link e ne attenua le statistiche in base a quanti link l'evento ha attraversato. Questa attenuazione serve ad evitare che un evento si propaghi indefinitamente, garantendo una sorta di "attenuazione" dell'impatto man mano che l'evento attraversa vari gruppi.

Un altro aspetto del processo di diffusione esterna è che, una volta che l'evento raggiunge il gruppo destinatario, il primo NPC che deve ricevere l'evento viene scelto casualmente per poi ripetere la diffusione all'interno del gruppo come descritto prima.

Integrazione dei Concetti Temporal

Un elemento essenziale per rendere il modello realistico è l'integrazione della componente temporale. Nella realtà, la propagazione degli eventi non avviene in maniera istantanea: c'è sempre un ritardo, un tempo necessario affinché l'evento si diffonda da una persona all'altra. Per modellare questo aspetto, il sistema introduce il concetto di **velocità di diffusione**, che calcola quanto tempo ci si mette a propagare un evento.

Per la **velocità di diffusione** degli eventi all'interno del gruppo, si considerano fattori analoghi a quelli che influenzano la probabilità di trasmissione, ma che vengono combinati per produrre un tasso di propagazione espresso in Unit per un certo intervallo. Le variabili che influenzano la velocità sono:

- **Grandezza del Gruppo (G):** Come già descritto, un gruppo più numeroso permette una trasmissione più rapida, in quanto vi sono più canali attraverso cui l'evento può diffondersi.
- **Affiatamento (A):** Un alto livello di coesione accelererà il processo di diffusione, in quanto i membri del gruppo reagiscono in maniera coordinata.
- **Importanza dell'Evento (I):** Eventi di maggiore importanza generano risposte più immediate e forti, traducendosi in una velocità di propagazione maggiore.
- **Contesto Storico (C):** Questo parametro modula la scala temporale della diffusione, traducendo il valore numerico in un intervallo temporale concreto. Il contesto storico definisce il ritmo comunicativo: in un ambiente con comunicazioni lente, una stessa combinazione numerica potrebbe tradursi in un tempo maggiore rispetto a contesti moderni.

La velocità di propagazione viene quindi calcolata con la seguente formula:

$$V_{interno} = G \cdot A \cdot I \cdot C + \varepsilon [Unit]$$

Qui:

- G: Questo funge da **base** per il calcolo della velocità.
- I, A, C: I loro valori agiscono come **moltiplicatori** che aumentano o diminuiscono la velocità di base e, i suoi valori, sono compresi tra 0 e 2.
- ε è un piccolo termine casuale aggiunto alla somma, il cui scopo è quello di introdurre una componente di casualità nella velocità di diffusione. Questo termine garantisce che, pur essendo il valore teorico calcolato basato sui parametri, la trasmissione reale avvenga con leggere variazioni, riflettendo l'imprevedibilità delle interazioni reali.

Per la diffusione esterna, invece, la velocità si calcola in modo analogo, ma utilizzando parametri specifici per il collegamento tra gruppi:

$$V_{esterno} = T \cdot D \cdot I \cdot C + \varepsilon [Unit]$$

dove:

- T è il contributo della tipologia del legame tra i gruppi e agisce da **moltiplicatore**.
- D rappresenta la distanza relazionale tra i gruppi e agisce da **base**.
- C, I: Come definiti in precedenza, agiscono da **moltiplicatori**.
- ε è un piccolo termine casuale che assicura che la velocità di diffusione non sia rigidamente costante, ma presenti leggere variazioni.

Queste formule per la velocità consentono di ottenere un valore numerico che rappresenta quante Unit di tempo sono necessarie affinché l'evento si diffonda all'interno di un gruppo o attraversi il confine verso un altro gruppo.

Unit

Per interpretare i valori ottenuti dalle formule di velocità è necessario definire un'unità di misura temporale, denominata Unit. Esistono due approcci principali per stabilire il valore di 1 Unit: l'approccio assoluto e quello variabile.

Con l'approccio assoluto il valore di 1 Unit è fisso e corrisponde a un intervallo di tempo predefinito, ad esempio, 1 giorno, 1 settimana o 1 mese, indipendentemente dal contesto. Questa soluzione, pur offrendo semplicità e uniformità nel confronto tra simulazioni, risulta limitata perché non tiene conto delle differenze nei ritmi comunicativi che caratterizzano contesti storici e ambientali diversi.

L'approccio variabile, invece, modula il valore di 1 Unit in base alle condizioni del contesto, come crisi sociali o naturali e, in particolare, il periodo storico considerato. Ad esempio, in epoche in cui la comunicazione procede più lentamente, 1 Unit potrà rappresentare un intervallo temporale più lungo, mentre in periodi moderni o durante situazioni di crisi che accelerano il flusso informativo, il valore si ridurrà per riflettere un ritmo più rapido.

Dopo aver valutato le due opzioni, si è scelto di adottare l'approccio variabile, basato esclusivamente sul periodo storico, per garantire una rappresentazione più realistica e coerente dei ritmi comunicativi nelle diverse epoche.

Periodo	Tempo
Antichità	1 Mese
Medioevo	1 Settimana
Rinascimento	3 Giorni
Era Industriale	1 Giorno
Epoca Moderna	1 Ora
Era Digitale	10 Minuti

Tabella 1: esempio di Unit variabile al contesto storico

Problemi con Questa Soluzione

L'attuale modello per la diffusione degli eventi si basa su due principi critici: la scelta casuale del prossimo NPC a ricevere l'evento e la trasmissione dell'evento inalterato lungo la catena comunicativa. Questi aspetti presentano diverse criticità che compromettono la capacità del sistema di replicare fedelmente le dinamiche reali di comunicazione.

- **Scelta Casuale del Prossimo NPC**

Come detto prima, la scelta del prossimo NPC a ricevere l'evento è effettuata in modo puramente casuale, ossia ogni NPC ha la stessa probabilità di ricevere l'evento, indipendentemente dalle relazioni esistenti o dalle preferenze personali. Questo approccio presuppone implicitamente che tutti gli NPC siano isolati e privi di connessioni, contraddicendo il comportamento osservato nella comunicazione umana.

Nella realtà, le informazioni vengono trasmesse principalmente a persone con cui si ha un legame o una certa familiarità. Amicizie, rapporti di fiducia e connessioni sociali influenzano notevolmente il flusso comunicativo: un individuo diffonde una notizia principalmente a chi conosce e con cui ha una relazione, non a sconosciuti. La scelta casuale, ignorando queste relazioni, porta a una trasmissione indiscriminata dell'evento, non consultando la rete sociale del trasmettitore.

Questo meccanismo riduce la capacità del modello di simulare accuratamente la "selezione sociale". In un sistema ideale, la probabilità di trasmettere l'evento dovrebbe aumentare se il ricevente ha una relazione consolidata con chi lo invia e diminuire in assenza di un legame significativo.

- **Impossibilità di Distorcere l'Evento durante la Propagazione**

Un ulteriore limite riguarda la mancanza di meccanismi per "distorcere" o modificare l'evento durante la sua propagazione. In una comunicazione naturale, ogni volta che un'informazione viene trasmessa, subisce una trasformazione: chi la riceve la interpreta in base alla propria personalità, esperienza e relazione con il mittente. Questo processo, spesso definito "effetto senza fili", comporta una variazione del contenuto man mano che l'evento viaggia da una persona all'altra.

Nel modello attuale l'evento viene propagato in modo invariato e questo perché non si sa quale NPC è stato a condividere l'evento. Questa mancanza significa che l'NPC dovrebbe distorcere l'evento

solo in base alla propria personalità oppure non distorcerlo affatto, ma questo porta far sì che, indipendentemente dalle caratteristiche dell'NPC ricevente, l'evento rimane identico.

La mancanza di una distorsione, oppure una incompleta, compromette il realismo della simulazione. Senza la possibilità di modificare adeguatamente il contenuto dell'evento, il sistema non riesce a catturare la diversità delle narrazioni che, nella vita reale, variano da persona a persona.

L'approccio casuale adottato nella scelta del prossimo NPC e l'assenza di un adeguato meccanismo di distorsione dell'evento sono due problemi critici che limitano significativamente la capacità del sistema di rappresentare realisticamente la diffusione delle informazioni in un contesto sociale. La casualità implica che ogni agente viene trattato in maniera identica, ignorando i legami esistenti, mentre l'impossibilità di modificare l'evento priva il modello della capacità di adattare il contenuto in base alla personalità e alla storia delle interazioni. Questi limiti, presi insieme, portano a una simulazione meno realistica e riducono la profondità e la variabilità che caratterizzano la comunicazione umana autentica.

3.4 Diffusione degli eventi: seconda idea

La prima idea di diffusione degli eventi, analizzata in precedenza, ha evidenziato numerosi limiti legati alla scelta casuale e alla mancanza di personalizzazione nella trasmissione degli eventi. Per ovviare a tali problematiche, è stata sviluppata una nuova soluzione che modifica il concetto di base andando verso la versione inizialmente presentata: anziché propagare gli eventi tramite link che collegano gruppi interi o in maniera casuale all'interno degli stessi, si crea una nuova mappa incentrata sui singoli NPC. In questa nuova struttura, i nodi rappresentano gli NPC individuali, mentre i link corrispondono alle relazioni esistenti tra di essi, indipendentemente dal gruppo di appartenenza.

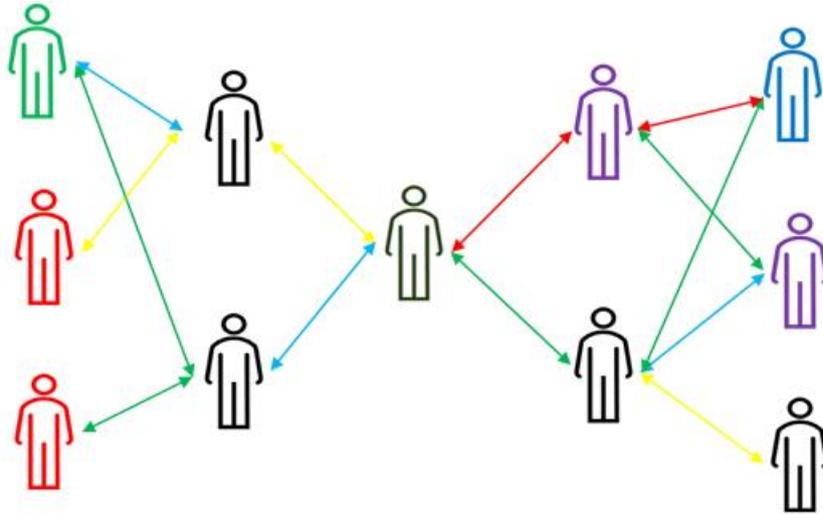


Figura 5: esempio di rete sociale

Questo approccio permette una diffusione più realistica, in cui la trasmissione dell'evento può essere distorta in base alla personalità dell'NPC e alla qualità della relazione con il trasmettitore. Di conseguenza, la rete di relazioni di ogni NPC comprende non solo individui dello stesso gruppo, ma anche appartenenti a gruppi differenti, rendendo la propagazione del messaggio molto più articolata e contestualizzata.

In questa nuova visione, la diffusione degli eventi si sviluppa tenendo in considerazione le caratteristiche individuali degli NPC, la natura dell'evento e soprattutto la relazione specifica tra chi trasmette e chi riceve l'informazione. Di seguito verranno illustrate in dettaglio le formule e i meccanismi che regolano la diffusione a livello individuale e tra gruppi, nonché il nuovo utilizzo dei link tra gruppi per la condivisione degli "Eventi Ufficiali".

Propagazione degli Eventi all'Interno del Gruppo e Velocità

Nella nuova mappa, la propagazione degli eventi a livello individuale si basa sulla combinazione di tre fattori principali: la personalità del diffusore, ovvero chi propaga l'evento, le caratteristiche intrinseche dell'evento e la relazione specifica che intercorre tra il diffusore e quello il ricevente del messaggio.

1. Il Fattore di Personalità

La personalità del diffusore viene quantificata attraverso una serie di attributi numerici, ciascuno compreso tra 0 e 1, che includono: espansività, impulsività, eloquenza, carisma. Questi parametri vengono combinati per ottenere un moltiplicatore, definito come:

$$F_{personality} = clamp(a \cdot extraversion + \beta \cdot impulsivity + \theta \cdot charisma - \gamma \cdot reliability, F_{min}, F_{max})$$

Qui α , β , θ e γ sono coefficienti che ponderano rispettivamente l'influenza di extraversion, impulsivity e charisma, mentre la reliability viene sottratta per penalizzare la diffusione in presenza di bassa affidabilità. I limiti F_{min} e F_{max} garantiscono che il valore finale rimanga entro un range prestabilito.

2. Il Fattore Evento

Il secondo elemento riguarda le caratteristiche intrinseche dell'evento stesso. Da qui viene ricavato un moltiplicatore, indicato come $K_{event}(typology, npc, peopleInvolved)$ che restituisce un valore compreso tra 0 e 2 che dipende dalla tipologia dell'evento e dalla relazione che il diffusore ha con le persone coinvolte.

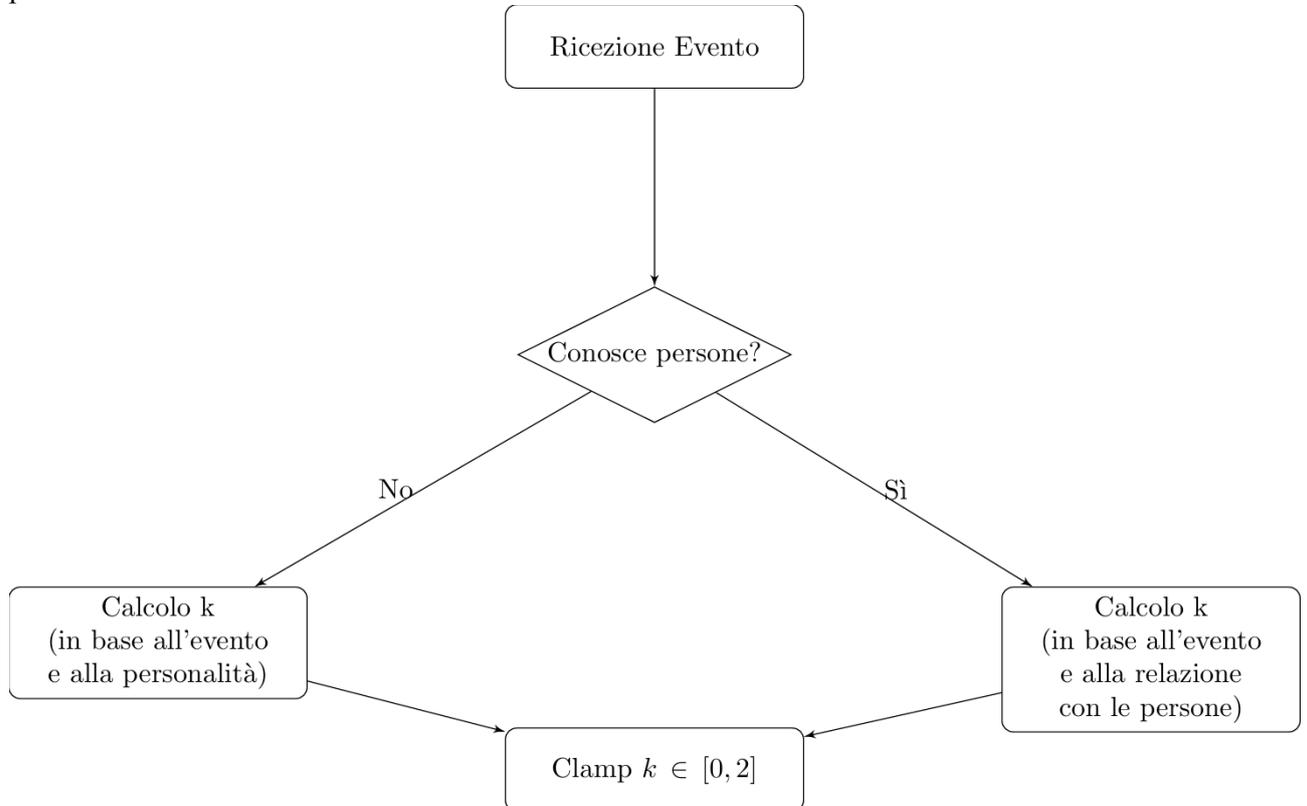


Figura 6: Semplice esempio di k

Accanto a questo moltiplicatore, si introduce il parametro P , definito come:

$$P = 1 - e^{-I \cdot T}$$

Dove:

- T rappresenta la tipologia dell'evento
- I la sua importanza.

La combinazione di P e K_{event} consente di adattare l'impatto dell'evento in funzione sia delle sue proprietà intrinseche sia del contesto relazionale.

3. La Relazione tra Diffusore e Ricevente

Il terzo fattore considerato è la relazione esistente tra il diffusore dell'evento e l'NPC che lo riceve. Questa relazione è misurata come un valore numerico, che rappresenta la forza del legame tra i due individui. Tale valore, compreso tra -1 e 1, viene integrato nel calcolo finale per modulare la probabilità che l'evento venga recepito e, di conseguenza, propagato ulteriormente.

Combinando questi fattori, la probabilità di diffusione dell'evento a livello individuale all'interno del gruppo è definita dalla seguente formula:

$$P_{interno}(receiver) = (P \cdot \alpha + rel[receiver] \cdot (1 - \alpha)) \cdot F_{personality} \cdot K_{event}$$

Velocità di Diffusione Interna

La velocità con cui l'evento si propaga tra i singoli NPC, espressa in unità temporali (Unit), viene definita dalla formula:

$$T(receiver) = (1 + \delta - |rel[receiver]|) \cdot (1 + \delta - I) \cdot H + \varepsilon [Unit]$$

Qui:

- $|rel[receiver]|$ rappresenta la forza della relazione tra il diffusore e il ricevente.
- I è l'importanza dell'evento.
- H è un fattore legato al contesto storico.
- δ è un margine introdotto per evitare che il valore della formula si annulli quando relazione e importanza raggiungono il massimo.
- ε è un piccolo termine casuale.

Questa formula garantisce che il tempo T sia inversamente proporzionale sia alla forza della relazione $|rel[receiver]|$ sia all'importanza dell'evento I , permettendo così agli eventi più rilevanti o condivisi tra individui vicini di diffondersi più rapidamente.

Diffusione degli Eventi all'Esterno del Gruppo e Velocità

Per la diffusione degli eventi tra gruppi, il modello sfrutta un meccanismo che tiene conto di fattori simili a quelli usati per la propagazione interna, ma con una componente aggiuntiva legata alla distanza tra i gruppi. In questo nuovo schema, la diffusione tra gruppi avviene attraverso la nuova mappa incentrata sui singoli NPC, ma è necessario considerare la distanza relazionale tra il gruppo del diffusore e quello del ricevente.

Per includere il contributo della distanza, si definisce un valore normalizzato, *normalizedD*, calcolato come:

$$normalizedD = 1 + \frac{\text{distanza tra i due gruppi}}{\text{distanza massima tra due gruppi nel sistema}}$$

Questo parametro consente di ridurre l'efficacia della trasmissione dell'evento quando la distanza relazionale tra i gruppi è elevata, poiché una maggiore distanza implica una comunicazione meno diretta e più filtrata.

Formula per la Propagazione Esterna

La probabilità di diffusione dell'evento all'esterno del gruppo è data dalla seguente formula:

$$P_{esterno}(receiver) = \left(p \cdot \alpha + \frac{rel[receiver]}{normalizedD} \cdot (1 - \alpha) \right) \cdot F_{personality} \cdot K_{event}$$

In questa espressione, il termine $\frac{rel[receiver]}{normalizedD}$ rappresenta l'effetto della relazione, attenuato dalla distanza normalizzata, mentre i coefficienti α e $(1 - \alpha)$ bilanciano l'influenza di P e della relazione. Questa formula consente di mantenere il modello coerente con la logica adottata per la diffusione interna, ma introduce la variabile distanza per rendere il passaggio dell'evento tra gruppi più realistico.

Velocità di Diffusione Esterna

La velocità con cui un evento attraversa il confine tra gruppi e si propaga nel gruppo ricevente è definita come:

$$T(receiver) = [(1 + \delta - |rel[receiver]|) \cdot D \cdot H] + \varepsilon [Unit]$$

Qui:

- $|rel[receiver]|$ è la forza della relazione tra il diffusore e il ricevente.
- D rappresenta la distanza tra i gruppi, non normalizzata in questa formula.
- H è il fattore contestuale storico.
- δ è un margine introdotto per evitare che il valore della formula si annulli quando relazione e importanza raggiungono il massimo.
- ε è il termine casuale che introduce variazioni minime nel tempo di diffusione.

Questo calcolo permette di determinare quanto tempo (in Unit) impiega l'evento a propagarsi da un gruppo all'altro, tenendo conto della distanza e della qualità della relazione.

I Link tra i Gruppi: Nuovo Utilizzo

Con la transizione al nuovo sistema di propagazione, incentrato sui singoli NPC, il ruolo dei link tra gruppi subisce una trasformazione significativa. Nel modello precedente, i link tra gruppi venivano utilizzati per propagare gli eventi in maniera globale, ora, con la nuova mappa in cui i nodi sono i singoli NPC e le relazioni vengono valutate individualmente, i link tra gruppi assumono una nuova funzione.

I link tra gruppi vengono riservati alla condivisione di una particolare tipologia di eventi, denominati "Eventi Ufficiali". Questi eventi ufficiali differiscono dagli eventi standard per alcune caratteristiche fondamentali:

- **Non Vengono Distorti:** Gli Eventi Ufficiali sono trasmessi in maniera invariata; non subiscono alcuna distorsione durante la propagazione. La descrizione e il contenuto dell'evento rimangono fedeli all'informazione originale.
- **Non Presentano Statistiche Specifiche:** A differenza degli eventi standard, che sono accompagnati da un insieme di statistiche derivanti dal contesto della conversazione, gli Eventi Ufficiali non vengono corredati da statistiche di impatto o intensità.
- **Nessuna Coinvolgimento Individuale:** Questi eventi non implicano la partecipazione attiva degli NPC, ma vengono distribuiti a livello di gruppo. La loro diffusione si basa esclusivamente sulla relazione esistente tra i gruppi.

La probabilità di diffusione di un Evento Ufficiale tra gruppi è calcolata come:

$$P_{diff}(j) = |rel[j]| \cdot I$$

dove $|rel[j]|$ rappresenta la forza della relazione tra il gruppo corrente e il gruppo j , e I indica l'importanza dell'evento.

Per quanto riguarda la velocità di diffusione degli Eventi Ufficiali, il processo è simile a quello per gli eventi standard. La velocità, dunque, si determina principalmente in base alla forza del collegamento tra i gruppi e alla distanza.

Infine, la scelta del primo NPC a cui viene condiviso un Evento Ufficiale all'interno di un gruppo avviene in base a criteri gerarchici: si seleziona l'NPC appartenente alla classe gerarchica più alta. Anche se la gerarchia non è ancora stata introdotta in dettaglio, questo meccanismo garantisce che la diffusione degli Eventi Ufficiali segua un percorso di alta affidabilità e rappresenti fedelmente le strutture organizzative presenti nel sistema.

Osservazioni Finali

La nuova idea di diffusione degli eventi, basata su una mappa individuale dei singoli NPC e sulle relazioni personali, rappresenta un notevole miglioramento rispetto al modello precedente. Integrando il fattore di personalità, la natura dell'evento e la relazione specifica tra diffusore e ricevente, il sistema consente una trasmissione più realistica e contestualizzata degli eventi. La possibilità di distorcere l'evento in base alle caratteristiche dell'NPC e al tipo di relazione introduce una variabilità essenziale, che rispecchia la complessità delle interazioni umane.

Il nuovo modello prevede meccanismi specifici per la diffusione sia all'interno dei gruppi che tra gruppi, con formule che tengono conto della distanza relazionale e dei fattori contestuali storici. L'introduzione dei link tra gruppi per la gestione degli Eventi Ufficiali fornisce un ulteriore livello di organizzazione, riservando una particolare modalità di diffusione a eventi che devono essere trasmessi in modo invariato e senza distorsioni.

Questo approccio, pur essendo più complesso e richiedendo un'accurata calibrazione dei parametri, come i coefficienti nella formula di personalità e la normalizzazione della distanza tra gruppi, offre un modello molto più aderente alla realtà. Il sistema, infatti, non assume più che ogni NPC sia isolato, ma integra la rete delle relazioni interpersonali in maniera dinamica, permettendo alla diffusione degli eventi di evolversi in base alle specifiche interazioni e alle caratteristiche individuali.

La capacità di influenzare la propagazione dell'evento grazie a moltiplicatori come $F_{personality}$ e K_{event} , insieme alla possibilità di alterare la velocità di diffusione mediante il termine ε e il fattore storico H , garantisce un'ampia flessibilità e realismo nel comportamento del sistema. Il nuovo utilizzo dei link tra gruppi per gestire gli Eventi Ufficiali assicura che le informazioni critiche possano essere trasmesse in modo sicuro e controllato, rispettando le gerarchie e le strutture organizzative esistenti.

In conclusione, la seconda idea di diffusione degli eventi, basata su una mappa individuale e su relazioni personalizzate, offre una soluzione avanzata per superare le limitazioni dell'idea precedente. Essa consente di catturare in maniera dettagliata le sfumature della comunicazione, permettendo una trasmissione dinamica e contestualizzata degli eventi.

3.5 Gli NPC: Descrizione e Interazione con gli LLM

3.5.1 Definizione e Strutturazione degli NPC

Nel sistema, gli NPC sono stati definiti come entità complesse e articolate, studiate per supportare scambi testuali coinvolgenti e coerenti. La struttura interna di ciascun NPC si basa su un insieme di attributi fondamentali che ne determinano il comportamento e la personalità, garantendo che ogni conversazione risulti sempre in linea con l'identità del personaggio. Gli attributi principali definiti sono:

- **Nome:** Un identificativo per ciascun NPC.
- **Età:** Un valore numerico che rappresenta l'età dell'NPC, utile per dare coerenza alla sua backstory e alle sue interazioni.
- **Gruppo di Appartenenza:** L'NPC è associato a un gruppo o una fazione, il che permette di collegare la sua storia e il suo comportamento a una comunità più ampia. Questo attributo è fondamentale anche per definire le relazioni intergruppo.
- **Backstory:** La storia personale dell'NPC, che ne fornisce il contesto, la motivazione e le esperienze pregresse. La backstory aiuta a rendere l'NPC più credibile e a giustificare i suoi comportamenti durante le conversazioni.
- **Ideologie:** I pensieri e le convinzioni dell'NPC riguardo alla vita, alla politica, e ad altri aspetti fondamentali. Questo attributo contribuisce a definire il suo punto di vista e a influenzare le sue risposte in conversazioni che toccano temi etici o sociali.
- **Situazione Familiare:** Lo stato relazionale dell'NPC, che include informazioni sullo status familiare e sulle eventuali relazioni importanti. Questo aspetto è utile per dare profondità alla personalità dell'NPC e per gestire le interazioni interpersonali.
- **NPC Conosciuti:** Un insieme che contiene i nomi degli altri NPC con cui l'NPC ha già interagito, accompagnata da un'indicazione della tipologia di relazione, ad esempio, amici, colleghi, rivali.
- **Personalità:** La personalità dell'NPC viene definita da una serie di valori numerici che vanno da 0 a 1. Gli attributi chiave sono:
 - affidabilità
 - espansività
 - impulsività
 - eloquenza
 - carisma
- **Mood:** Lo stato d'animo attuale dell'NPC è descritto da attributi quali:
 - felicità
 - rabbia
 - stress
- **Pensiero sul Giocatore:** Rappresenta la percezione che l'NPC ha del giocatore. È strutturato con attributi:
 - fiducia
 - notorietà
 - rispetto
- **Pensiero sul Gruppo:** Questo attributo descrive come l'NPC percepisce gli altri gruppi. Include variabili quali:
 - paura
 - fiducia
 - ostilità
 - cooperazione

Aggiornamento del Pensiero sul Giocatore e sul Gruppo

Uno degli aspetti dinamici e fondamentali del modello è l'aggiornamento continuo del "pensiero sul giocatore" e del "pensiero sul gruppo" in seguito a eventi che coinvolgono l'NPC. Questi aggiornamenti determinano come l'NPC adatta le sue percezioni in base alle interazioni, influenzando di conseguenza il comportamento nelle conversazioni future.

- **Pensiero sul Giocatore**

Il "pensiero sul giocatore" è strutturato attorno ad attributi quali fiducia, notorietà e rispetto. Quando l'NPC riceve un evento in cui il giocatore è coinvolto, questo pensiero viene aggiornato attraverso una formula che integra tre fattori principali:

1. **BaseDelta:**

Questo componente cattura l'effetto diretto dell'evento sulle emozioni dell'NPC, basandosi sulle statistiche dell'evento, in particolare sulle componenti di happiness, anger e stress. Ad esempio, per l'aggiornamento del parametro di fiducia, si utilizza una formula del tipo:

$$baseTrustDelta = event.happiness \cdot y_h(I) - event.anger \cdot y_a(I) - event.stress \cdot y_s(I)$$

Qui, $y_h(I)$, $y_a(I)$, $y_s(I)$ sono coefficienti che variano in funzione dell'importanza dell'evento, I . L'idea è che un evento caratterizzato da alta felicità aumenti la fiducia, mentre alti livelli di rabbia e stress la riducano. Questa formula è stata scelta perché permette di combinare in maniera bilanciata le componenti emotive dell'evento, fornendo una base quantitativa per l'aggiornamento.

2. **PersonalityFactor:**

Questo fattore modula l'impatto dell'evento sulla percezione del giocatore in base alla personalità intrinseca dell'NPC. Ad esempio, se un NPC è naturalmente incline a fidarsi degli altri, l'effetto positivo dell'evento sarà amplificato. La formula usata per il trust è:

$$trustFactor = 1 + trust \cdot y_t$$

Dove y_t è un coefficiente che determina quanto l'atteggiamento intrinseco dell'NPC influenzi l'aggiornamento della fiducia. L'idea è quella di fornire un moltiplicatore che aumenti o diminuisca l'effetto base in funzione della predisposizione dell'NPC.

3. **EventFactor:**

Infine, l'aggiornamento viene ulteriormente modificato da un fattore che tiene conto degli attributi specifici dell'evento. Questo "eventFactor" può essere definito come:

$$eventFactor = conf(per) \cdot typology(per) \cdot |relationship|$$

Qui, $conf(per)$ rappresenta un moltiplicatore legato alla confidenzialità dell'evento e a come l'NPC reagisce quando lo riceve, mentre $typology(per)$ riflette la tipologia dell'evento nella prospettiva dell'NPC. Il termine $|relationship|$ indica la forza della relazione esistente tra l'NPC attuale e quello coinvolto nell'evento. Questo fattore è fondamentale perché permette di personalizzare ulteriormente l'aggiornamento in base al contesto relazionale, evidenziando come la natura dell'evento interagisca con la relazione preesistente.

L'aggiornamento finale del pensiero sul giocatore per un determinato attributo, ad esempio la fiducia, è il risultato della combinazione di questi tre fattori. La formula complessiva, in forma semplificata, può essere espressa come:

$$\Delta trustPlayer = baseTrustDelta \cdot trustFactor \cdot eventFactor$$

Questo approccio consente di modellare l'effetto di ogni evento in modo granulare, assicurando che l'aggiornamento del pensiero sul giocatore rifletta sia le emozioni suscitate dall'evento, sia la predisposizione dell'NPC e la natura della relazione interpersonale.

Pensiero sul Gruppo

Il "pensiero sul gruppo" viene aggiornato in maniera analoga, utilizzando una struttura che considera variabili quali paura, fiducia, ostilità e cooperazione. Anche qui, l'aggiornamento avviene tramite un processo che integra:

- **BaseDelta:** Calcolato sulla base delle statistiche dell'evento che influenzano la percezione dell'NPC nei confronti di un determinato gruppo.
- **PersonalityFactor:** Che modula l'effetto in base alle inclinazioni dell'NPC, ad esempio la sua propensione alla cooperazione o all'ostilità.
- **EventFactor:** Che tiene conto della tipologia e della confidenzialità dell'evento e della relazione che l'NPC con coloro che coinvolti nell'evento.

La logica è identica a quella applicata al pensiero sul giocatore, con la differenza che qui il focus è sul gruppo. L'aggiornamento per un attributo del pensiero sul gruppo, ad esempio la fiducia verso il gruppo, si esprime con una formula simile:

$$AtrustGroup = baseTrustDelta \cdot trustFactor \cdot eventFactor$$

Dove ogni componente viene definita analogamente, ma con parametri che riflettono il contesto intergruppo. L'adozione di queste formule consente al sistema di aggiornare dinamicamente le percezioni degli NPC, rendendo le interazioni più fluide e contestuali, e garantendo che ogni evento abbia un impatto calibrato sia sul pensiero verso il giocatore che su quello verso gli altri gruppi.

File di Configurazione per il Comportamento degli NPC

Per garantire una maggiore variabilità e realismo nelle interazioni, il sistema prevede un file di configurazione che permette di personalizzare il comportamento degli NPC. Questo file di configurazione è strutturato in tre parti principali: **Actions**, **Behavior** e **Behavior Event**. Ognuna di queste sezioni ha uno scopo ben definito e viene utilizzata per regolare come l'NPC risponde in situazioni diverse.

- **Actions**

```
ACTION nameAction {
  if (condition) {
    shortTermBehavior = "short term behavior"
    conversationHint = "conversation hint"
    conversationStatus = "conversation status"
    personalized = "personalized field" [optional]
  } else {
    shortTermBehavior = "short term behavior"
    conversationHint = "conversation hint"
    conversationStatus = "conversation status"
    personalized = "personalized field" [optional]
  }
}
```

La sezione Actions è progettata per definire le risposte immediate e contestuali degli NPC in base a specifiche condizioni. Ogni ACTION è un blocco di codice che stabilisce, per una determinata situazione, quale comportamento l’NPC debba adottare. La struttura tipica di un’azione è la seguente:

Ad esempio, un’azione come "greetUser" potrebbe essere configurata per far sì che l’NPC saluti il giocatore con entusiasmo se il suo mood è positivo e il pensiero sul giocatore è favorevole. Le variabili shortTermBehavior, conversationHint e conversationStatus sono utilizzate per guidare la conversazione in maniera dinamica, mentre il campo personalized può essere utilizzato per ulteriori personalizzazioni specifiche dell’NPC.

Questa sezione consente di impostare comportamenti immediati e contestuali, offrendo una base flessibile per le interazioni quotidiane dell’NPC.

- **Behavior**

La sezione Behavior si occupa di definire flussi di conversazione più strutturati, che regolano come l’NPC deve rispondere a determinati argomenti sollevati dal giocatore. Per farlo, si è introdotta una struttura chiamata “flow”, che associa ad un particolare argomento un set di esempi sia per il giocatore che per l’NPC. La struttura segue questo schema:

```
define flow <keyword-for-argument>:
```

```
    user <examples-identifier>
```

```
    npc <examples-identifier>
```

```
define user <examples-identifier>:
```

```
    "example phrase n.1"
```

```
    "example phrase n.2"
```

```
    "example phrase n.3"
```

```
    "example phrase n.4"
```

```
define npc <examples-identifier>:
```

```
    if condition:
```

```
        "example phrase n.1"
```

```
    if condition:
```

```
        "example phrase n.2"
```

Questa struttura consente di creare un modello flessibile e modulare per le conversazioni. In pratica, per ogni argomento di interesse (ad esempio, "talk about work"), viene definito un flow che include esempi di frasi che il giocatore potrebbe utilizzare per introdurre l’argomento e risposte che l’NPC dovrebbe fornire, in base alle condizioni attuali.

Le condizioni presenti nel blocco if permettono di personalizzare la risposta in funzione di parametri dinamici come il suo stato d’animo e il pensiero sul giocatore, garantendo che la conversazione rimanga sempre contestuale e personalizzata.

- **Behavior Event**

La terza parte del file di configurazione è il Behavior Event, che regola come l’NPC deve reagire dopo aver ricevuto un evento. Questo modulo definisce la reazione dell’NPC in modo specifico, determinando il tipo di risposta, il commento associato ed eventuali campi personalizzati. La struttura tipica è la seguente:

```
if (condition) {  
    reaction = "reaction type"  
    commentary = "commentary"  
    personalized = "personalized field" [optional]  
} else {  
    reaction = "reaction type"  
    commentary = "commentary"  
    personalized = "personalized field" [optional]  
}
```

Ad esempio, se l’NPC riceve un evento di tipo "POSITIVE_NEWS", e le sue condizioni interne sono soddisfatte, il Behavior Event determina che l’NPC reagisca accogliendo la notizia con gioia, diffondendola ulteriormente nella sua rete.

Le condizioni in questo blocco possono includere parametri come gli NPC conosciuti, lo stato d’animo, il pensiero sul giocatore e sugli altri gruppi, che insieme definiscono in maniera precisa come l’NPC deve comportarsi in seguito a un evento.

Le sezioni Actions, Behavior e Behavior Event del file di configurazione sono progettate per garantire che il comportamento degli NPC sia modulabile e personalizzabile in base alle esigenze narrative e contestuali. Il file di configurazione permette ai creatori di definire le risposte degli NPC in maniera dettagliata, specificando non solo le azioni immediate da compiere ma anche i flussi conversazionali e le reazioni agli eventi.

3.5.2 Creazione di Nuovi NPC

L'obiettivo è quello di consentire la creazione di nuovi NPC in modo semplice e flessibile, nonostante la loro complessità. Poiché gli NPC sono caratterizzati da numerosi attributi che li descrivono, dalla personalità alla backstory, dalle ideologie alla situazione familiare, il sistema impone un solo vincolo obbligatorio: il gruppo di appartenenza. Tutti gli altri campi sono opzionali e, se omessi, verranno generati automaticamente dal sistema. Questa generazione automatica avviene mediante una serie di richieste sequenziali al modello IA, in modo tale che le informazioni siano prodotte in un ordine logico. In particolare, i campi come: gli NPC conosciuti, gli eventi noti e il pensiero sul giocatore e sugli altri gruppi sono calcolati mediante meccanismi dedicati, che tengono conto delle relazioni e delle affinità tra i personaggi.

Generazione degli Attributi

Per tutti gli attributi che non vengono calcolati se il creatore non fornisce questi dati, il sistema li genera automaticamente tramite richieste al modello IA. Data la sequenzialità delle richieste l'ordine delle richieste è cruciale:

1. Nome
2. Personalità
3. Ideologie
4. Famiglia
5. Storia personale
6. Gli NPC conosciuti
7. Lo stato d'animo
8. Il pensiero sul giocatore e sui gruppi

Il modello integra in ogni richiesta tutte le informazioni precedenti, in modo che la richiesta relativa alla storia personale includa dettagli sulla personalità, le ideologie e la famiglia, generando risposte coerenti con il contesto informativo accumulato. Oltre a questo, riceve come input non soltanto i dati specifici dell'NPC, ma anche una descrizione completa del mondo e dell'ambiente in cui vive. Grazie a questa metodologia, il risultato finale consiste in un set di attributi armonizzati e consistenti, in grado di definire in maniera accurata l'NPC e di riflettere fedelmente l'evoluzione del contesto comunicativo e narrativo, che rendono il sistema estremamente dinamico.

Calcolo degli Attributi

La parte più complessa della creazione di un nuovo NPC riguarda il calcolo dei valori derivati. Questi calcoli permettono di impostare la probabilità che due NPC si conoscano e, successivamente, di definire il "pensiero sul giocatore" e il "pensiero sugli altri gruppi" sulla base di questi collegamenti.

- **Generazione della Lista degli NPC Conosciuti**

La lista degli NPC conosciuti viene generata in base all'affinità stimata. Il sistema esamina, per ciascun NPC già presente, se realizzare un nuovo collegamento dopo aver calcolato la probabilità che il nuovo NPC sia correlato ad un altro. Se la probabilità risulta favorevole, il nuovo NPC viene inserito nella lista dei conoscenti del primo e viceversa. In seguito, per ogni NPC già incluso, il sistema avvia una procedura per diffondere gli eventi passati. In pratica, per ogni evento noto ad un membro della lista, viene calcolata la probabilità che anche il nuovo lo conosca. Se tale probabilità si realizza, l'evento viene aggiunto alla lista degli eventi conosciuti dal nuovo NPC. Questa procedura consente di rendere il nuovo personaggio parte integrante della rete di comunicazione e interazioni preesistenti.

Il calcolo della probabilità deriva dall'affinità tra l'NPC che si vuole aggiungere alla lista e ciascuno di quelli già esistenti. L'idea alla base è che due NPC abbiano maggiori possibilità di conoscersi se le loro personalità risultano compatibili. Per farlo, i parametri della personalità vengono suddivisi in tre gruppi concettuali:

- **Similarità:** Questo gruppo include attributi come affidabilità ed espansività. La similarità viene calcolata per ogni parametro come:

$$C_t = 1 - |t_1 - t_2|$$

Dove t_1 e t_2 sono i valori dell'attributo per i due NPC. Questo calcolo esprime quanto i due valori siano vicini; un valore prossimo a 1 indica alta similarità, mentre un valore più basso indica differenza significativa.

- **Positività:** Comprende gli attributi relativi al lato più "positivo" della personalità, come eloquenza. Il contributo di positività viene calcolato come:

$$C_t = \frac{t_1 - t_2}{2}$$

In questo modo, viene ottenuta una media dei valori, che riflette il grado di positività condiviso dai due NPC.

- **Negatività:** Questo gruppo si concentra su attributi che potrebbero influire negativamente, come l'impulsività. Qui, il contributo viene calcolato come:

$$C_t = 1 + \frac{t_1 + t_2}{2}$$

Tale formula assicura che, se entrambi gli NPC mostrano alti livelli di impulsività, il valore risultante sarà basso, segnalando una bassa affinità in questo aspetto.

Una volta ottenuti i contributi per ogni gruppo, l'affinità complessiva a viene calcolata come la somma pesata di ciascun contributo:

$$a = \sum \omega_\tau \cdot C_\tau [0,1]$$

Dove

- ω_τ rappresenta il peso assegnato a ciascun gruppo
- C_τ è il contributo calcolato per quel gruppo.

Ma la probabilità che due NPC si conoscano non dipenda solamente dall'affinità individuale, ma anche da un ulteriore fattore: l'affiatamento del gruppo. L'affiatamento è un valore compreso tra 0 e 2 che funge da moltiplicatore. Per NPC appartenenti allo stesso gruppo, la probabilità finale si calcola come:

$$P = a \cdot fellowship$$

Nel caso in cui gli NPC appartengano a gruppi diversi, l'affinità viene ulteriormente modulata dalla relazione esistente tra i gruppi. In questo scenario, la probabilità che due NPC si conoscano si calcola come:

$$P = \beta \cdot a + (1 - \beta) \cdot relationship[j]$$

dove $relationship[j]$ è il valore della relazione tra il gruppo del nuovo NPC e il gruppo del secondo NPC, e β è un coefficiente che bilancia l'influenza dell'affinità individuale rispetto a quella intergruppo. Questa formula consente di aumentare la probabilità se i gruppi hanno una relazione positiva o di diminuirla in caso contrario.

- **Calcolo del Pensiero sul Giocatore e sugli Altri Gruppi**

Una volta che l'affinità e le relazioni sono state stabilite, il sistema procede alla generazione del "pensiero sul giocatore" per il nuovo NPC. La generazione di questo pensiero avviene in due fasi: prima viene valutato se l'NPC possiede un pensiero "eccezionale" e successivamente viene calcolato il valore finale, integrando le influenze derivanti dagli eventi.

Questo concetto di pensiero "eccezionale" si basa sul confronto tra le proprietà dell'NPC e quelle corrispondenti del gruppo di riferimento. Per ogni proprietà considerata, il sistema, calcola la differenza rispetto al valore medio del gruppo. Se la differenza supera una soglia predefinita nel senso opposto, cioè se il valore dell'NPC è **significativamente** diverso da quello medio del gruppo, quella proprietà è considerata "eccedente". Affinché il pensiero complessivo possa essere classificato come eccezionale, almeno l'80% delle proprietà valutate devono superare tale soglia.

Ad esempio, con una soglia fissata a 0.3, se il valore di fiducia dell'NPC è 0.6 e il valore medio del gruppo è 0.2, la differenza di 0.4 viene considerata valida; al contrario, se il valore è 0.7 e il gruppo ha un valore di 1.0, la differenza, pur essendo 0.3, non è considerata valida in quanto non rispetta la direzione richiesta. Il sistema aggrega il numero di proprietà "eccedenti" e, confrontandolo con il totale delle proprietà considerate, determina se il pensiero del giocatore dell'NPC può essere classificato come eccezionale.

Indipendentemente dal fatto che il pensiero risulti eccezionale o meno, il valore finale del pensiero sul giocatore viene calcolato con la formula:

$$TP = \gamma \cdot TP_{known} + (1 - \gamma) \cdot TP_{events}$$

Dove:

- TP_{known} è il pensiero derivato dal confronto con il pensiero dei suoi conoscenti. Per ogni NPC conosciuto, si calcola un indice di relazione *relIndex* definito come:

$$relIndex = 2 \cdot relationship[knwonNpc] - 1$$

e un valore C calcolato come:

$$C = \frac{determination + charisma - impulsivity}{3}$$

Il valore C indica il grado di indipendenza di giudizio; un valore alto suggerisce maggiore autonomia, mentre un valore basso indica una maggiore tendenza a conformarsi al pensiero del gruppo.

- TP_{events} viene invece calcolato in base alla personalità e agli eventi che coinvolgono il giocatore, attraverso una richiesta al modello IA che tiene conto del contesto del mondo in cui l'NPC vive.
- γ è un coefficiente che bilancia il contributo del pensiero derivato dai conoscenti e quello generato dagli eventi.

Se, per esigenze narrative, l'NPC deve avere un pensiero "eccezionale" sul giocatore ma il valore TP risultante non lo è, viene applicata una correzione:

$$TP = TP_{group} + \gamma(TP - TP_{group})$$

dove TP rappresenta il valore del pensiero e TP_{group} il riferimento medio del gruppo, mentre γ è un fattore di amplificazione iterativo (maggiore di uno) che spinge il valore verso l'estremo desiderato.

Il "pensiero sugli altri gruppi" viene calcolato seguendo la stessa logica, applicando le formule di aggiornamento e bilanciamento per riflettere come l'NPC percepisce gruppi differenti rispetto al proprio.

3.5.3 Comportamento degli NPC alla Ricezione degli Eventi

Quando un NPC riceve un evento, il sistema applica un processo articolato in più fasi, pensato per simulare in maniera realistica il modo in cui le informazioni vengono elaborate, interpretate e propagate nella vita reale. Questo processo, che va dal filtraggio iniziale passa dall'aggiornamento della relazione con gli NPC coinvolti, fino ad arrivare alla distorsione dell'evento e all'aggiornamento dei pensieri, permette di gestire la complessità delle interazioni interpersonali, tenendo conto sia delle caratteristiche intrinseche del messaggio sia delle relazioni esistenti tra il diffusore e il ricevente.

Fase di Filtraggio dell'Evento

La prima fase nella ricezione di un evento è il filtraggio. Alla creazione dell'NPC vengono definite delle regole e dei vincoli che determinano quali eventi debbano essere considerati e quali, invece, debbano essere scartati. Le decisioni del filtro sono basate sia sull'evento che sugli altri parametri dell'NPC: per ogni evento, si può valutare se l'informazione è rilevante, se riguarda persone a cui l'NPC tiene e tanti altri aspetti. Se l'evento non soddisfa questi criteri, viene semplicemente ignorato, evitando così che informazioni non pertinenti o troppo deboli interferiscano con il flusso comunicativo.

Questo meccanismo di filtraggio è essenziale per prevenire il sovraccarico di informazioni. In una situazione reale, non tutte le informazioni vengono tenute in considerazione: spesso ignoriamo dettagli minori o eventi che riteniamo irrilevanti per lo stato emotivo o per il contesto in cui ci troviamo. Il filtro, quindi, agisce da prima linea di difesa, selezionando solo quegli eventi che hanno il potenziale di influenzare l'NPC.

Aggiornamento delle relazioni

Se l'evento supera il filtro, si procede all'aggiornamento delle relazioni tra l'NPC designato come C e i soggetti coinvolti nell'evento, identificati come A e B. In questo contesto, la ricezione dell'evento determina un aggiornamento reciproco delle relazioni, espresso dalle seguenti formule:

$$\begin{aligned}\Delta R_{CA} &= F_{event} \cdot \eta \cdot R_{CB} \\ \Delta R_{CB} &= F_{event} \cdot \eta \cdot R_{CA}\end{aligned}$$

Dove F_{event} rappresenta l'effetto diretto dell'evento ed è definito come:

$$F_{event} = I \cdot T \cdot F_{personality}$$

Con:

- I il contributo dell'importanza dell'evento, ed è compreso tra [-1, 1]
- T il contributo della tipologia dell'evento
- $F_{personality}$ come definito in precedenza

Se R_{CA} è positivo, l'evento tende a rafforzare ulteriormente la relazione o a penalizzarla in modo più marcato se l'evento è negativo; se il rapporto è neutro allora l'aggiornamento dipenderà solo dal valore diretto.

Successivamente, il nuovo valore della relazione viene calcolato sommando l'aggiornamento al valore precedente, così da ottenere:

$$R_{CA}^{New} = R_{CA}^{Old} + \Delta R_{CA} \text{ e } R_{CB}^{New} = R_{CB}^{Old} + \Delta R_{CB}$$

In questo modo, se C ha già una relazione forte con B, questo influisce positivamente, o negativamente, sul modo in cui C aggiorna il legame con A, e viceversa.

La stessa logica si applica anche nei casi in cui l’NPC C conosce solamente uno dei soggetti coinvolti o, al contrario, non ha alcuna relazione preesistente con nessuno dei due. Ad esempio, se C conosce unicamente A, l’aggiornamento si semplifica in:

$$\Delta R_{CA} = F_{event}$$

Poiché, in questo scenario, R_{CB} è pari a 0. Analogamente, nel caso in cui C non conosca né A né B, l’aggiornamento per entrambe le relazioni verrà determinato esclusivamente dal valore diretto dell’evento, ovvero:

$$\Delta R_{CA} = F_{event} \quad \Delta R_{CB} = F_{event}$$

Distorsione dell'Evento

Il passo successivo è la distorsione dello stesso. La distorsione degli eventi nella vita reale si riferisce al fenomeno per cui le informazioni, man mano che vengono trasmesse da una persona all'altra, subiscono modifiche dovute a interpretazioni soggettive, bias cognitivi e perdite o alterazioni dovute al passaggio attraverso la memoria e la comunicazione. In altre parole, l'evento originale viene "trasformato" a ogni passaggio, in modo che il messaggio finale possa differire dalla fonte originale.

Nel sistema, la distorsione dell'evento è modellata considerando tre aspetti fondamentali:

1. Interpretazione Soggettiva:

Ogni NPC interpreta un evento in base alla propria personalità, alla propria backstory e all’esperienza accumulata. Questi fattori determinano come l'informazione viene modificata internamente. Ad esempio, un NPC con una personalità molto ottimista potrebbe enfatizzare gli aspetti positivi di un evento, mentre uno più cinico potrebbe concentrarsi sui suoi aspetti negativi. Questa interpretazione soggettiva permette di trasformare l'evento in modo che rispecchi il punto di vista personale del ricevente. La trasformazione inizia applicando i valori della personalità già definiti, modificando la descrizione dell’evento per renderla coerente con l’identità dell’NPC.

2. Bias Cognitivi:

I pregiudizi e le aspettative personali giocano un ruolo significativo nella modulazione dell'informazione. Il sistema tiene conto del “pensiero sul giocatore” o sul gruppo, influenzando così la ricezione del messaggio. Se l'NPC nutre una percezione favorevole, tende a esaltare gli aspetti positivi dell'evento; al contrario, una visione negativa porta a una interpretazione più critica e selettiva. Questi bias agiscono come ulteriori filtri, non solo modificando il contenuto, ma anche la rilevanza dell'informazione in base al grado di fiducia o sospetto nei confronti dei soggetti coinvolti.

3. Effetto “Telefono Senza Fili”:

Infine, l’evento subisce ulteriori modifiche a causa delle imperfezioni intrinseche alla comunicazione, simili al noto gioco del “telefono senza fili”. Ad ogni passaggio, piccole imprecisioni si accumulano, contribuendo a una trasformazione progressiva del messaggio originario. Questo processo evidenzia come, anche in assenza di intenzioni malevoli, il contenuto possa alterarsi in modo significativo durante la sua diffusione.

La distorsione dell'evento nel sistema simula fedelmente il modo in cui, nella vita reale, la trasmissione delle informazioni viene influenzata dalla soggettività, dai pregiudizi e dalle inevitabili imperfezioni comunicative.

Ecco un esempio che mostra come lo stesso evento possa essere raccontato in due modi differenti:

- **Versione Originale:**

" Questa mattina, mentre attraversavo la piazza del mercato, ho notato Marina che mi osservava con il solito sguardo carico di antipatia. Ignorandola, mi sono diretta al banco delle stoffe per esaminare alcuni tessuti destinati a un nuovo vestito. Non appena ho scelto una stoffa che mi piaceva, Marina si è avvicinata rapidamente e, con un gesto deliberatamente brusco, me l'ha strappata di mano. Con voce alta e piena di malizia ha esclamato: «Francesca, davvero credi che un abito fatto con questa stoffa possa migliorare il tuo aspetto? Nemmeno il più abile dei sarti potrebbe riuscirci!»
I presenti attorno a noi hanno cominciato a ridacchiare apertamente. Umiliata e indignata, ho abbassato lo sguardo, sentendo le guance avvampare di vergogna. Marina mi ha fissata con un sorriso soddisfatto, evidentemente compiaciuta della mia umiliazione, prima di allontanarsi lasciandomi sola in mezzo alla piazza, amareggiata e profondamente frustrata per aver subito, ancora una volta, la sua crudele provocazione."

- **Versione Distorta:**

" Questa mattina, attraversando la piazza del mercato, Marina ha dimostrato ancora una volta la sua schiettezza e prontezza di spirito. Nel suo tipico modo diretto e senza fronzoli, ha intercettato Francesca mentre quest'ultima stava scegliendo un tessuto, offrendole un'opinione chiara e sincera sulla scelta. Senza alcuna cattiveria, Marina ha voluto sottolineare la scarsa qualità del tessuto, evitando che Francesca si trovasse a indossare qualcosa che non l'avrebbe valorizzata. Anche se Francesca ha reagito visibilmente scossa, è evidente che Marina, pur con il suo approccio risoluto, abbia agito con l'intento di essere d'aiuto, mostrando un interesse genuino nel garantire che Francesca non commettesse un errore nell'acquisto. Nonostante la reazione del pubblico, che ha trovato divertente la scena, è evidente che la preoccupazione di Marina per l'immagine e la presentabilità di Francesca sia stata sincera e ben intenzionata."

- **Spiegazione del Contesto:**

La versione originale dell'evento è stata raccontata da Francesca, che ha vissuto direttamente l'accaduto, a Rodolfo ma, il rapporto tra i due, è segnato da reciproca antipatia. Al contrario, Rodolfo ha un profondo legame di amicizia con Marina. Questa differenza nel valore delle relazioni ha influenzato profondamente l'interpretazione di Rodolfo: ascoltando il racconto di Francesca, lui ha filtrato e rielaborato i fatti attraverso la sua simpatia per Marina, trasformando un gesto che in realtà appare umiliante in un'azione diretta e, per lui, persino utile.

Trigger dell'Aggiornamento dei Pensieri

Una volta distorto, l'evento non rimane semplicemente come informazione ma innesca l'aggiornamento dei "pensieri sul giocatore" e del "pensiero sul gruppo" dell'NPC. Se l'evento coinvolge il giocatore, il sistema utilizza il messaggio distorto per aggiornare i parametri di fiducia, notorietà e rispetto, seguendo le formule descritte in precedenza. Analogamente, se l'evento riguarda un NPC di un gruppo, il pensiero sull'altro gruppo viene modificato in base al contenuto dell'evento. Questo aggiornamento è fondamentale per garantire che le percezioni dell'NPC evolvano in maniera dinamica, riflettendo l'impatto degli eventi recenti e adattandosi alle nuove informazioni. Il meccanismo di trigger è integrato nel flusso di ricezione, in modo che ogni evento elaborato contribuisca in modo incrementale alla revisione delle percezioni, influenzando le interazioni future.

Controllo della Similarità tra Eventi

Un ulteriore controllo critico riguarda la gestione della similarità degli eventi. Durante la sua vita è possibile che l'NPC riceva eventi che presentano contenuti simili, anche se provenienti da fonti differenti. Per evitare la ridondanza e garantire che il personaggio gestisca le informazioni in modo coerente, è necessario confrontare i contenuti degli eventi per determinare se siano sostanzialmente identici.

Per questo compito, sono state considerate due possibili soluzioni:

- **Richiesta al modello IA:**

Questo approccio sfrutta le capacità interpretative di un modello di intelligenza artificiale per confrontare le descrizioni testuali degli eventi. In pratica, il modello IA riceve in input le descrizioni di due eventi e il contesto in cui sono avvenute e, basandosi su una serie di criteri interni, restituisce una misura di similarità.

- **Uso delle tecniche di NLP – Cosine Similarity:**

In questo approccio, le descrizioni degli eventi vengono trasformate in embedding, ovvero rappresentazioni vettoriali in uno spazio multidimensionale, utilizzando modelli di embedding testuale consolidati. Una volta ottenuti i vettori, si applica la formula classica della cosine similarity:

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} [-1, 1]$$

dove A e B sono i vettori che rappresentano le descrizioni degli eventi. Il valore risultante, che varia tipicamente tra -1 e 1, quantifica l'angolo tra i due vettori: un valore vicino a 1 indica una forte similarità, mentre valori inferiori suggeriscono una minore somiglianza.

Questo metodo si concentra esclusivamente sul contenuto testuale degli eventi, escludendo il contesto ambientale o narrativo, il quale in questo caso non incide in modo rilevante sulla valutazione della similarità.

La decisione sull'approccio più adatto per il controllo della similarità tra eventi dipende da diverse considerazioni. Da un lato, l'impiego di tecniche NLP come la cosine similarity offre vantaggi significativi in termini di leggerezza computazionale e rapidità: grazie alla trasformazione delle descrizioni in embedding vettoriali, è possibile effettuare confronti diretti sul contenuto testuale, ottenendo misure di similarità rapide e ben definite. Questo metodo si focalizza esclusivamente sul significato delle parole e delle frasi, riducendo al minimo l'influenza del contesto ambientale o narrativo, il che lo rende particolarmente efficace in scenari dove sono richiesti numerosi confronti in tempo reale.

Dall'altro lato, la richiesta diretta al modello IA sfrutta le capacità interpretative e contestuali del sistema, potendo considerare non solo il contenuto testuale ma anche il contesto in cui gli eventi si sono verificati. Tale approccio, pur essendo più complesso e dispendioso in termini di risorse, può risultare prezioso in situazioni dove le sfumature narrative e i dettagli contestuali rivestono un'importanza maggiore per la corretta interpretazione degli eventi.

La scelta finale si basa sull'osservazione che, per valutare se due eventi siano simili, non è necessario considerare il contesto in cui la conversazione si è svolta, ma possiamo affidarci esclusivamente alla descrizione testuale dell'evento stesso. In altre parole, il confronto tra le informazioni si concentra sul contenuto specifico delle parole e delle frasi che compongono l'evento, escludendo elementi contestuali che, in questo caso, hanno un impatto trascurabile sulla valutazione della similarità. Questo approccio, che sfrutta tecniche di NLP come la cosine similarity, permette di ottenere una misura accurata e diretta del grado di somiglianza tra gli eventi.

Una volta stabilito che due eventi fanno riferimento allo stesso avvenimento, l'NPC deve decidere quale di essi conservare. Tale decisione si fonda sulla valutazione del livello del rapporto con la fonte di ciascun evento. In pratica, se il collegamento con il diffusore dell'evento attuale risulta più solido e affidabile rispetto a quello dell'evento già memorizzato, quest'ultimo verrà sostituito; altrimenti, il nuovo evento verrà scartato, mantenendo intatta l'informazione precedentemente acquisita.

Propagazione dell'Evento

Dopo che l'evento è stato filtrato, distorto e utilizzato per aggiornare i pensieri, esso viene infine propagato ulteriormente nel sistema. La propagazione avviene in base a meccanismi già descritti nei capitoli precedenti, ma il focus qui è sul fatto che l'evento, una volta elaborato, diventa il seme per ulteriori interazioni, influenzando il comportamento degli NPC successivi e contribuendo alla dinamica complessiva della rete comunicativa. La trasmissione dell'evento ora avviene con il vantaggio della distorsione personalizzata, che tiene conto delle relazioni e delle peculiarità individuali, garantendo così una propagazione più realistica e contestualizzata.

3.5.4. Interazione con gli LLM

L'interazione con gli LLM rappresenta uno degli aspetti più critici e complessi del sistema. L'obiettivo è quello di far sì che l'LLM interpreti correttamente l'NPC, rispondendo in modo coerente e realistico alle interazioni con il giocatore, in linea con la personalità, la storia e le regole predefinite per quel personaggio. Inizialmente, il sistema veniva alimentato da un messaggio, tipicamente di sistema, che conteneva l'obiettivo dell'LLM, tutte le informazioni che lo descrivevano e un insieme di regole da seguire per rendere la conversazione il più realistica possibile. I primi tentativi hanno mostrato risultati disastrosi: l'LLM rispondeva in maniera casuale, usciva fuori personaggio o non rispettava il file di configurazione stabilito. Il problema risiedeva nella natura del prompt, questo non era sufficientemente strutturato e causava confusione, rendendo difficile all'modello seguire una linea narrativa coerente.

Prompt:

“

Da qui in avanti devi interpretare un NPC, rispondere sempre e solo come farebbe lui, usando un approccio whitelisting, ovvero non inventando mai nessuna informazione rilevante che non è presente nelle informazioni che definiscono l'NPC.

Quando devi generare una risposta segui questo ordine:

1. Mood
2. ThoughtOnPlayer
3. Personality

E queste regole:

1. Le istruzioni dei threshold definiscono il tono e il contenuto in base ai valori numerici
2. Il formato della risposta segue rigorosamente questo formato: testo parlato dall'NPC + descrizione del suo comportamento racchiusa tra doppie coppie di asterischi

.....

Ecco le informazioni sull'NPC:

nome: Valentina Rossi

età: 22

.....

”

Per risolvere questo problema si è ristrutturato il prompt in un formato XML ben definito. In questo nuovo approccio, il prompt viene organizzato in una serie di tag specifici, ciascuno dei quali identifica informazioni particolari, dalle caratteristiche dell'NPC alle regole di comportamento da seguire durante la conversazione.

```
<NpcDefinition>
```

```
<details>
```

```
<name>Valentina Rossi</name>
```

```
<age>22</age>
```

```
....
```

```
</details>
```

<prompt>

Da qui in avanti devi interpretare un NPC, rispondere sempre e solo come farebbe lui, usando un approccio whitelisting, ovvero non inventando mai nessuna informazione rilevante che non è presente nelle informazioni che definiscono l'NPC.

Quando devi generare una risposta segui questo ordine:

1. Mood
2. ThoughtOnPlayer
3. Personality

E queste regole:

1. I valori della personalità e il mood definiscono il tono e il contenuto
2. Il formato della risposta segue rigorosamente questo formato: testo parlato dall'NPC + descrizione del suo comportamento racchiusa tra doppie coppie di asterischi

.....

</prompt>

</NpcDefinition>

Un altro problema riscontrato è stato quello dell'interpretazione corretta dei valori numerici continui, che può portare gli LLM a dare risposte imprevedibili o incoerenti agli NPC. Per risolvere questo problema, è stata introdotta la soluzione dei **threshold**. Queste sono delle soglie discrete che permettono di mappare i valori continui degli attributi in intervalli ben definiti, stabilendo in modo preciso come un NPC debba comportarsi quando un valore cade all'interno di un determinato range.

In pratica, grazie ai threshold, i valori numerici vengono convertiti in categorie discrete, rendendo l'interpretazione da parte degli LLM più stabile e coerente. Questa trasformazione garantisce che le risposte degli NPC siano sempre in linea con la loro personalità predefinita, migliorando così la consistenza e l'affidabilità dell'interazione con il sistema.

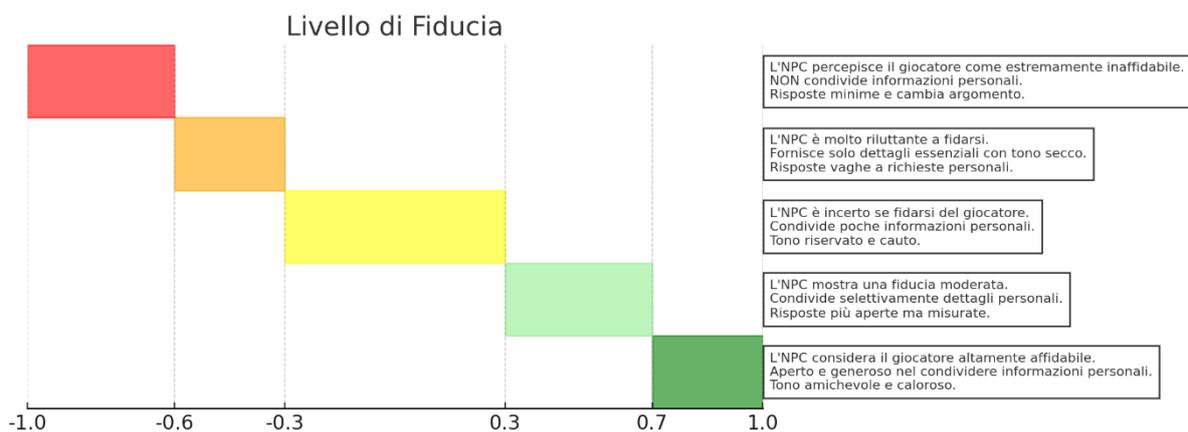


Figura 7: esempio di threshold del pensiero sul giocatore

L'uso dell'XML e dei threshold ha portato a notevoli miglioramenti: l'LLM ha iniziato a rimanere in personaggio, seguire con maggiore aderenza le regole predefinite e rispettare in parte il file di configurazione. Nonostante questo progresso, si sono riscontrate ancora alcuni problemi: alcune risposte sono state generate in maniera casuale, non tutte le regole sono state applicate correttamente e, in certi casi, l'LLM non ha rispettato del tutto il file di configurazione.

Per risolvere definitivamente tali problematiche, è stato introdotto il concetto del "pre-prompt". Il pre-prompt è un messaggio che precede l'XML e contiene una descrizione esaustiva e dettagliata di tutte le regole da seguire. In questo messaggio vengono spiegati come interpretare ogni tag, il flusso da seguire e le regole comportamentali che l'LLM deve rispettare. Questa soluzione si è rivelata ottimale: l'LLM ha iniziato a rispondere correttamente in tutte le possibili condizioni, mantenendo la coerenza e restando fedelmente in personaggio.

Nel corso dei test, però, si è notato che all'aumentare della lunghezza del prompt, comune in ambienti con un contesto complesso, il problema non scompare del tutto. Con modelli LLM di dimensioni ridotte, il prompt molto lungo rischia di superare il contesto massimo gestibile, mentre modelli più grandi, pur non superando il limite, tendono a mostrare errori: alcune risposte vengono generate casualmente oppure alcune regole vengono ignorate. Questi problemi evidenziavano come, con l'aumentare della complessità e della quantità di informazioni fornite, il modello si confonde e perde precisione.

RTGS System

Per affrontare questo problema, viene introdotto un sistema denominato RTGS, o *Attorney System*. RTGS è l'acronimo di:

- **Rule Engine**
- **Topic Recognition**
- **Information Gathering**
- **Information Selection**

Questo sistema ha l'obiettivo principale di ridurre la dimensione del prompt fornito all'LLM, mantenendo però intatta la coerenza del personaggio. L'RTGS consente di filtrare e selezionare soltanto le informazioni strettamente necessarie per generare una risposta coerente, eliminando dati superflui e riducendo così la quantità di testo che l'LLM deve elaborare in ogni singola richiesta.

Entrando nel dettaglio di ogni componente, abbiamo che:

- **Rule Engine**

Il Rule Engine è il componente responsabile di verificare, alla luce della conversazione in corso e di un set predefinito di regole, quali regole si applicano. In pratica, agisce come un classificatore che analizza se alcune regole si possono applicare al flusso della conversazione, ad esempio, se il giocatore abbia insultato l'NPC, se l'input sia incoerente o privo di senso (Nonsense Input Rule) o se il giocatore stia facendo troppe domande (Too Many Questions Rule). Il riconoscimento di queste regole è gestito tramite una macchina a stati, che monitora continuamente l'andamento della conversazione e mantiene traccia delle regole attive.

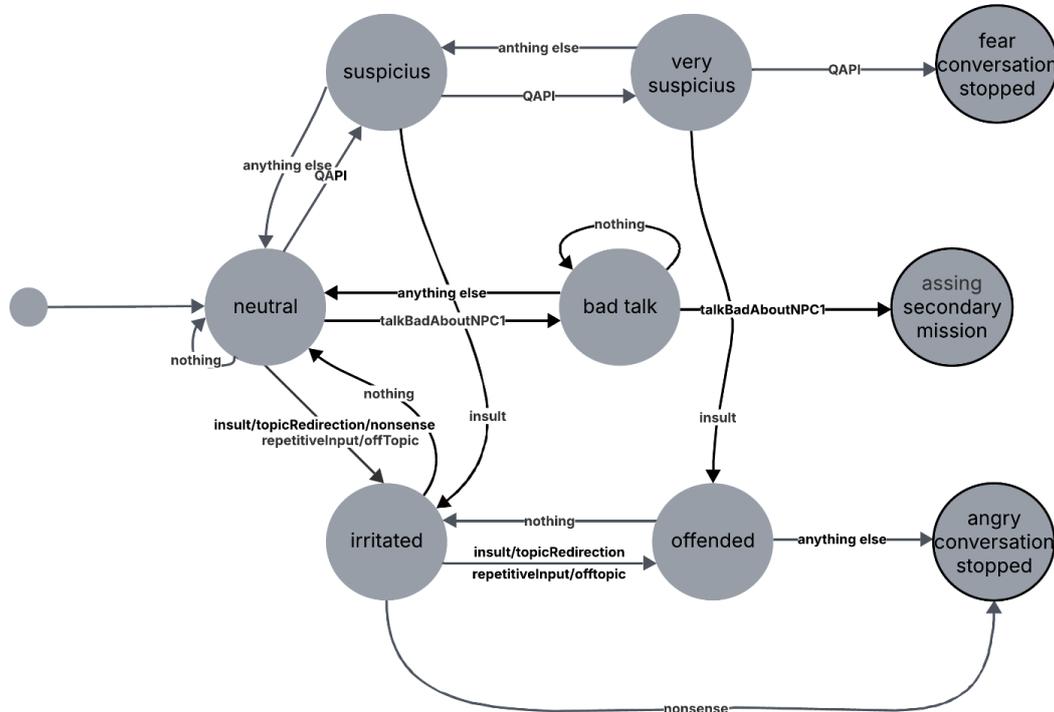


Figura 8: semplice macchina a stati finiti

- **Topic Recognition**

Il componente di Topic Recognition ha il compito di identificare il tema centrale della conversazione, attribuendo ad esso delle etichette e dei sottotemi. In pratica, agisce come un classificatore che assegna, sulla base della conversazione in corso, con particolare enfasi sull'ultima frase del giocatore, una coppia di etichette: topic e subtopic. Ad esempio, le etichette possono includere:

- **Family:** con subtopic come madre, padre, fratelli/sorelle, figli, coniuge.
- **Context:** che comprende informazioni sul mondo in cui vive l'NPC, con subtopic: globale, locale, corrente o meta.
- **Details:** che fornisce informazioni personali sull'NPC.
- **Events:** che descrive gli eventi ricevuti dall'NPC, con sottotemi come i nomi degli NPC coinvolti.
- **Thought on Other Groups:** che esprime le opinioni dell'NPC sugli altri gruppi, con il nome del gruppo come sottotema.

Questa componente garantisce che il sistema sappia esattamente quale argomento sta emergendo e, di conseguenza, quale insieme di informazioni è rilevante per la generazione della risposta successiva.

- **Information Gathering**

Una volta identificato il topic della conversazione, il sistema procede con l'Information Gathering. Lo scopo di questa fase è quello di recuperare tutte le informazioni necessarie relative all'argomento in corso. Il sistema raccoglie dati di base sull'NPC, come nome, backstory, personalità e altri attributi, insieme alle informazioni specifiche legate al topic riconosciuto. Questa raccolta di informazioni assicura che l'LLM disponga di tutto il contesto necessario per generare una risposta coerente e ben informata.

- **Information Selection**

Infine, l'Information Selection ha il compito di filtrare e selezionare solamente le informazioni essenziali, eliminando dati superflui che potrebbero ingombrare il prompt. Ad esempio, per la personalità dell'NPC, anziché trasmettere tutti i valori numerici viene selezionata soltanto il giusto threshold per ciascun attributo. Questa operazione permette di ridurre drasticamente la dimensione del prompt senza perdere informazioni critiche, garantendo che l'LLM riceva solo i dati strettamente necessari per mantenere la coerenza e l'aderenza al personaggio.

Il seguente diagramma di flusso mostra il percorso completo della generazione della risposta, dall'input iniziale dell'utente fino alla risposta finale dell'NPC. L'NPC sfrutta l'RTGS per generare un prompt ottimizzato, che viene inoltrato all'LLM, ottenendo così una risposta coerente e contestualizzata.

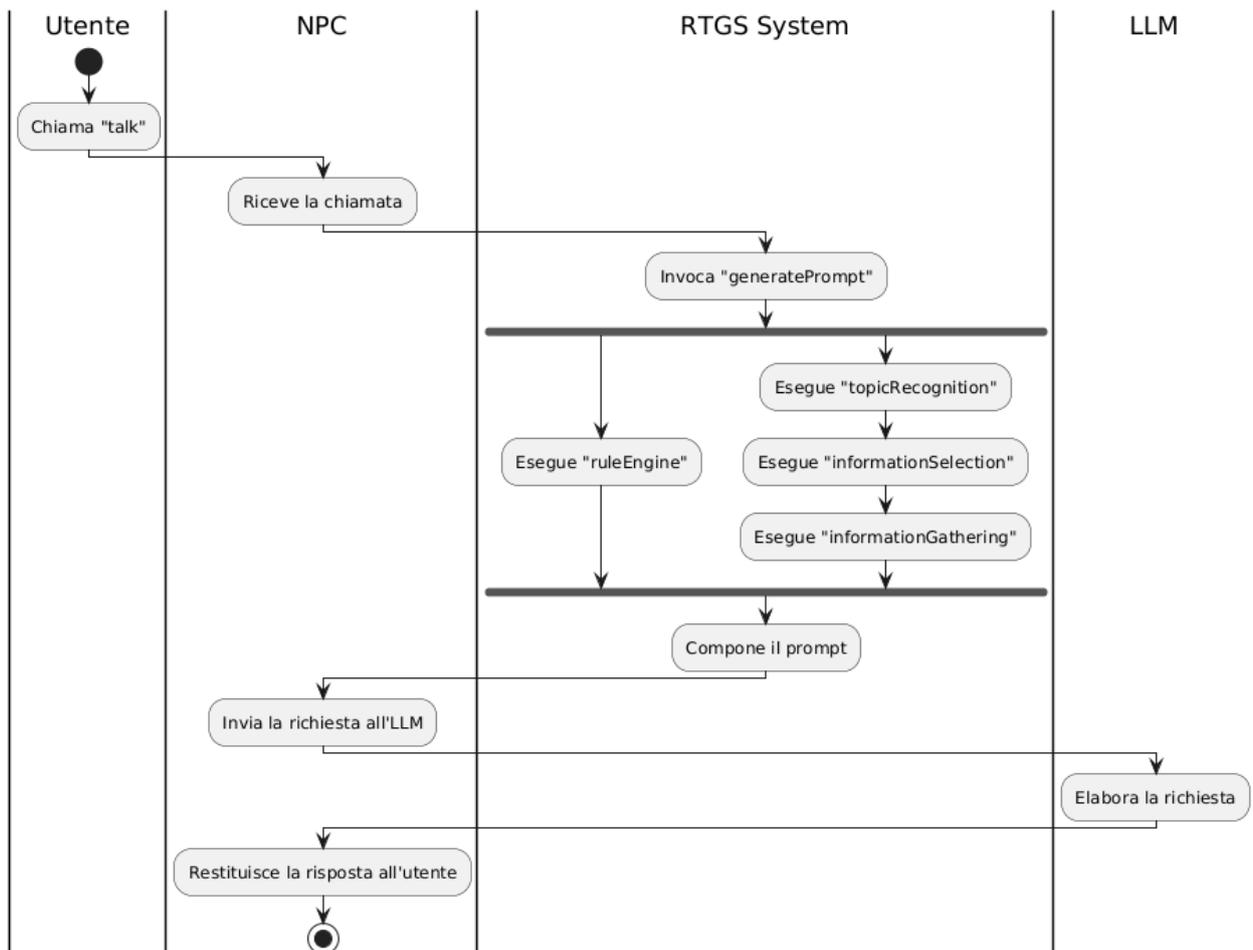


Figura 9: diagramma di flusso dell'RTGS System

Analisi delle Performance del RTGS System

Uno dei principali problemi che si riscontrano nel RTGS System riguarda le performance: il processo di estrazione delle informazioni, la classificazione dei topic e il riconoscimento delle regole richiede numerose richieste di classificazione al modello. Dato il numero di richieste da parte del Topic Recognition e del Rule Engine, il tempo di attesa prima di ricevere una risposta può risultare abbastanza alto.

In una prima implementazione, il Topic Recognition effettua tre richieste distinte, una per il topic, una per il sub-topic e una per verificare quali parti del file di configurazione vengono prese in considerazione, mentre il Rule Engine ne esegue due, una prima e una successiva alla risposta dell'NPC. Questo schema porta a tempi medi di attesa che oscillano tra i 10 e i 20 secondi, una situazione che può essere ulteriormente aggravata se le richieste non vengono effettuate in locale, ma via rete.

Per migliorare l'efficienza, in una seconda implementazione si adotta un approccio che prevede richieste sincrone, laddove possibile, e una riduzione complessiva del numero di chiamate, integrando alcune funzioni, come topic e sub-topic, in un'unica richiesta. Questa soluzione permette di dimezzare i tempi di attesa, riducendoli a un range compreso tra i 5 e i 10 secondi, pur rimanendo sensibili alle condizioni del network in caso di richieste online.

Composizione del Prompt

La composizione del prompt rappresenta un aspetto cruciale per permettere all'LLM di interpretare correttamente l'NPC e rispondere in maniera coerente. Il prompt finale che viene passato all'LLM è il risultato della fusione di diverse componenti. In particolare, il prompt è strutturato in quattro parti principali:

1. **Il Pre-Prompt:**

Il pre-prompt è un messaggio introduttivo che viene inviato prima del prompt principale con lo scopo di spiegare, in maniera esaustiva, come utilizzare ogni elemento del messaggio che descrive l'NPC.

2. **Descrizione l'NPC:**

La seconda componente del prompt è costituita da un messaggio strutturato, in questo caso XML, che descrive in dettaglio l'NPC.

3. **Le Regole per la Generazione delle Risposte:**

La terza parte del prompt include un insieme di regole e istruzioni che devono influenzare la generazione delle risposte dell'NPC. Queste regole derivano dal file di configurazione e dalla macchina a stati definita nel Rule Engine. In questo modo, l'LLM ha a disposizione tutte le informazioni necessarie per replicare il comportamento previsto e mantenere la coerenza narrativa.

4. **La Conversazione Precedente:**

Infine, il prompt include anche tutta la conversazione avvenuta fino a quel momento con l'NPC. Questa parte serve a fornire il contesto immediato e a garantire la continuità del dialogo aiutandolo a collegare risposte ed a evitare contraddizioni.

Questa composizione modulare consente di mantenere il prompt organizzato e facile da gestire ma permette anche al modello di interpretarlo al meglio.

3.5.5 Gestione della Memoria

Nel sistema, fino a questo momento, ogni volta che il giocatore terminava una conversazione con un NPC, l'intera interazione veniva cancellata e trasformata in un semplice evento memorizzato, senza alcuna conservazione dei dettagli specifici. Questo approccio, sebbene semplice, presenta evidenti limitazioni: se il giocatore riprende la conversazione poco tempo dopo, l'NPC si ritrova ad aver "dimenticato" dei piccoli dettagli che avevano caratterizzato l'interazione precedente, contrariamente a ciò che accade nella realtà. Per affrontare questo problema, è stato introdotto un sistema di gestione della memoria ispirato ai concetti di memoria a breve e a lungo termine, fenomeni fondamentali nel funzionamento cognitivo umano.

Nella psicologia cognitiva, la memoria è suddivisa in due categorie principali: la memoria a breve termine e la memoria a lungo termine. La memoria a breve termine permette di conservare temporaneamente le informazioni, mantenendo in mente i dettagli immediatamente rilevanti per la conversazione o per il compito corrente. Al contrario, la memoria a lungo termine si occupa di conservare le informazioni essenziali e riassunte per periodi prolungati, garantendo la continuità e la coerenza nel tempo.

Applicando questi concetti al sistema, si è sviluppato un meccanismo articolato in tre componenti principali:

1. **Registro della Memoria:**

Questo componente è suddiviso in due segmenti distinti: la memoria a breve e a lungo termine. Nel segmento di breve durata vengono conservati tutti gli scambi completi che avvengono tra l'NPC e il giocatore. Poiché lo spazio di memoria in questa sezione è limitato, la quantità di informazioni che può essere mantenuta è vincolata; ciò simula il fenomeno umano in cui la nostra memoria a breve termine ha una capacità finita. Al contrario, nel segmento di lunga durata vengono archiviate le informazioni essenziali, ottenute sintetizzando i dettagli delle conversazioni passate. Quando si raggiunge la dimensione massima oppure certe condizioni specifiche vengono soddisfatte, i dettagli minori vengono riassunti e trasferiti in questa sezione per garantire che l'NPC mantenga una continuità a lungo termine delle interazioni.

2. **Sintesi della Memoria:**

Quando il segmento di memoria a breve termine raggiunge il suo limite oppure vengono soddisfatte determinate condizioni, il sistema attiva il processo di sintesi. Questo processo prende le conversazioni archiviate nella memoria a breve termine, le sintetizza e le trasforma in una versione condensata, riassumendo le informazioni essenziali. Il meccanismo di sintesi simula il processo di "dimenticanza" presente nella memoria umana, in cui i dettagli meno importanti vengono scartati e solo le informazioni più rilevanti vengono conservate. Questo processo non solo riduce la dimensione delle informazioni memorizzate, ma assicura anche che l'NPC possa mantenere un ricordo coerente e aggiornato delle interazioni passate.

3. **Recupero della Memoria:**

Quando il giocatore fa riferimento a un evento o a un ricordo, il sistema attiva il processo di recupero. In questa fase, vengono utilizzate tecniche di NLP, in particolare la cosine similarity, per confrontare la frase del giocatore con gli embedding delle conversazioni riassunte presenti nella memoria a lungo termine. Se la similarità supera una soglia predefinita, il sistema identifica quel ricordo come rilevante. Nel caso in cui più ricordi superino la soglia, il sistema utilizza informazioni extra, come l'importanza del ricordo o quando è avvenuto, per determinare quale ricordo restituire al giocatore. In questo modo, l'NPC è in grado di fare riferimento a dettagli passati in maniera coerente, migliorando la continuità e la fluidità delle conversazioni.

Registro della Memoria

Il sistema di gestione della memoria utilizza un "Registro della Memoria" che integra due compartimenti distinti: quello a breve termine e quello a lungo termine.

- **Memoria a Breve Termine**

In questa sezione vengono conservate le conversazioni per intero. Non si salva semplicemente una sequenza continua di frasi, ma ogni conversazione viene archiviata come un oggetto individuale, comprensivo di tutte le informazioni rilevanti. Ogni oggetto conversazionale include:

- Il testo completo della conversazione.
- Un timestamp che indica il momento in cui la conversazione è avvenuta.
- Un parametro di importanza, derivato dal sistema di generazione degli eventi, che rappresenta la rilevanza emotiva o narrativa dell'interazione.

Questo registro è destinato a mantenere le informazioni per un breve periodo, sufficienti a garantire che, se il giocatore riprende la conversazione in un lasso di tempo ravvicinato, l'NPC possa richiamare dettagli specifici dell'interazione precedente.

- **Memoria a Lungo Termine**

Una volta che il Registro della Memoria a Breve Termine raggiunge il limite di capacità, o quando specifiche condizioni vengono soddisfatte, ad esempio è trascorso troppo tempo dall'interazione, il sistema attiva il processo di sintesi per trasferire i dati essenziali. Gli oggetti in memoria a lungo termine contengono:

- Un riassunto della conversazione, che condensa i punti salienti in un testo sintetico.
- Un timestamp, che indica il momento in cui il riassunto è stato creato.
- Il parametro di importanza della conversazione riassunta.

Questo meccanismo consente all'NPC di mantenere una continuità nelle interazioni a lungo termine, ricordando i dettagli essenziali senza dover gestire un enorme volume di testo.

Sintesi della Memoria

La "Sintesi della Memoria" è il processo che entra in funzione quando il Registro della Memoria a Breve Termine raggiunge il limite della sua capacità oppure quando si verificano determinate condizioni specifiche. Alcuni esempi di queste condizioni includono il tempo trascorso dalla conversazione, in quanto dopo alcuni giorni è naturale dimenticare i dettagli più precisi, oppure la comparsa di eventi più importanti che richiedono spazio nella memoria attiva.

Durante questo processo, il sistema invia una richiesta al modello IA per sintetizzare il contenuto della conversazione, utilizzando tecniche analoghe a quelle impiegate per generare descrizioni sintetiche degli eventi. Il risultato è una versione condensata che mantiene gli elementi fondamentali del dialogo, pur riducendo la quantità complessiva di informazioni da conservare.

La sintesi non ha soltanto lo scopo di ridurre il carico informativo, ma serve anche a preservare la coerenza narrativa a lungo termine, consentendo all'NPC di recuperare successivamente soltanto i ricordi essenziali della conversazione. Questo processo avviene automaticamente e in sequenza, garantendo che i dati trasferiti nella memoria permanente siano accuratamente integrati e fedeli ai contenuti originali.

Recupero della Memoria

Il processo di "Recupero della memoria" entra in gioco quando il giocatore fa riferimento a un ricordo o a un evento specifico che si trova nel Registro della Memoria a Lungo Termine. Per identificare il ricordo corretto, il sistema utilizza tecniche di NLP basate sulla *cosine similarity*. Le descrizioni riassunte vengono confrontate con la frase del giocatore e, se il valore ottenuto supera una soglia predefinita, il ricordo viene considerato rilevante.

Nel caso in cui più ricordi superino la soglia, il sistema impiega ulteriori criteri per la selezione. In particolare, si dà priorità al ricordo con il parametro di importanza più elevato; se vi sono ricordi con importanza simile, si seleziona quello più recente. Questo meccanismo di selezione garantisce che l’NPC faccia riferimento alla versione dell’evento più significativa e attuale, mantenendo così una continuità e una coerenza narrativa nell’interazione. Se il ricordo scelto non è quello corretto allora tutti i ricordi che superino la soglia, ad eccezione di quelli scelto in precedenza, verranno caricati in memoria in modo tale che l’LLM abbia le informazioni necessarie per poter capire quale ricordo utilizzare.

Analisi Comparativa con il paper Memory Repository for AI NPC

Questo capitolo analizza le principali differenze tra il modello sviluppato e descritto nella presente tesi e quello proposto nel paper "MemoryRepository for AI NPC" [13]. Soltanto dopo aver formalizzato e implementato il sistema descritto nell’elaborato, è emerso che una proposta simile era già stata affrontata in letteratura. Vengono dunque evidenziate in dettaglio le differenze metodologiche tra i due approcci e i relativi vantaggi che derivano dalla soluzione adottata nel presente lavoro.

I due modelli condividono concetti di base come la suddivisione della memoria in breve e lungo termine, il meccanismo di sintesi delle informazioni e l’utilizzo di tecniche NLP per il recupero dei ricordi. Tuttavia, sono presenti differenze rilevanti che comportano benefici sostanziali per il modello sviluppato nella tesi.

1. Archiviazione delle Conversazioni

- **Metodologia:** Nel sistema descritto in questo elaborato, ogni interazione viene memorizzata come un oggetto conversazionale completo, dotato di timestamp, parametro di importanza e struttura dati dedicata per una gestione individuale. Questo approccio permette una conservazione dettagliata delle interazioni, facilitando una successiva analisi più approfondita e precisa.
- **Confronto con il Paper:** Il paper, pur presentando un sistema basato su compartimenti distinti per la memoria, enfatizza maggiormente la capacità sintetica delle informazioni, dedicando meno attenzione alla granularità dell’archiviazione.
- **Vantaggio:** La maggiore precisione del sistema proposto assicura una sintesi più accurata grazie all’abbondanza di informazioni contestuali disponibili, permettendo una selezione più mirata delle informazioni rilevanti.

2. Recupero della Memoria e Ruolo del Rule Engine

- **Meccanismo di Recupero:** Il modello sviluppato combina tecniche NLP con il Rule Engine che determina, in base alla domanda, se recuperare informazioni dalla memoria a breve o a lungo termine. Nel caso di recupero dalla memoria a lungo termine si usano anche il valore dell’importanza e il timestamp nel caso in cui ci fossero più ricordi candidati.
- **Confronto con il Paper:** Il paper utilizza prevalentemente il confronto tramite la cosine similarity per identificare il ricordo più pertinente.
- **Vantaggio:** L’integrazione del Rule Engine consente al sistema di effettuare recuperi più contestualizzati e precisi, aumentando la coerenza narrativa e la continuità della conversazione.

Integrazione dei Concetti e Flusso Operativo

L'intero meccanismo di gestione della memoria è progettato per essere integrato nel sistema complessivo di interazione con gli NPC, in modo che la memoria a breve e a lungo termine possano interagire dinamicamente con gli altri moduli, come l'RTGS System. Quando il giocatore interagisce con un NPC, il sistema recupera automaticamente le informazioni memorizzate, includendo le conversazioni sintetizzate nel prompt del modello IA, in modo che la risposta tenga conto non solo dell'interazione attuale, ma anche della storia passata. Grazie a questa integrazione, è possibile delegare completamente la gestione degli eventi relativi al giocatore al sistema di memoria, liberando il prompt da questi dati.

Il meccanismo opera secondo un flusso ben definito:

1. Verifica delle Condizioni per il Riassunto:

L'RTGS controlla se sono soddisfatte le condizioni necessarie per attivare il meccanismo di sintesi. Ad esempio, verifica il tempo trascorso dalla conversazione più vecchia presente nel registro della memoria a breve termine oppure se si supera il limite del contesto gestibile. In tali situazioni, il sistema attiva il processo di riassunto, condensando le conversazioni in riassunti trasferiti poi nella memoria a lungo termine.

2. Verifica del Ricordo tramite il Rule Engine:

Il Rule Engine applica una regola specifica per determinare se l'evento o la conversazione a cui il giocatore fa riferimento sia presente nella memoria a breve o a lungo termine. Se il ricordo non viene individuato nella memoria a breve, il sistema attiva il meccanismo di recupero per estrarlo dalla memoria a lungo termine.

3. Integrazione nel Prompt di Conversazione:

Le informazioni più recenti, conservate nella memoria a breve termine, vengono direttamente inserite nella sezione della conversazione del prompt, mentre il ricordo recuperato dalla memoria a lungo termine viene trattato come un evento specifico del giocatore, garantendo così una risposta coerente e contestualizzata.

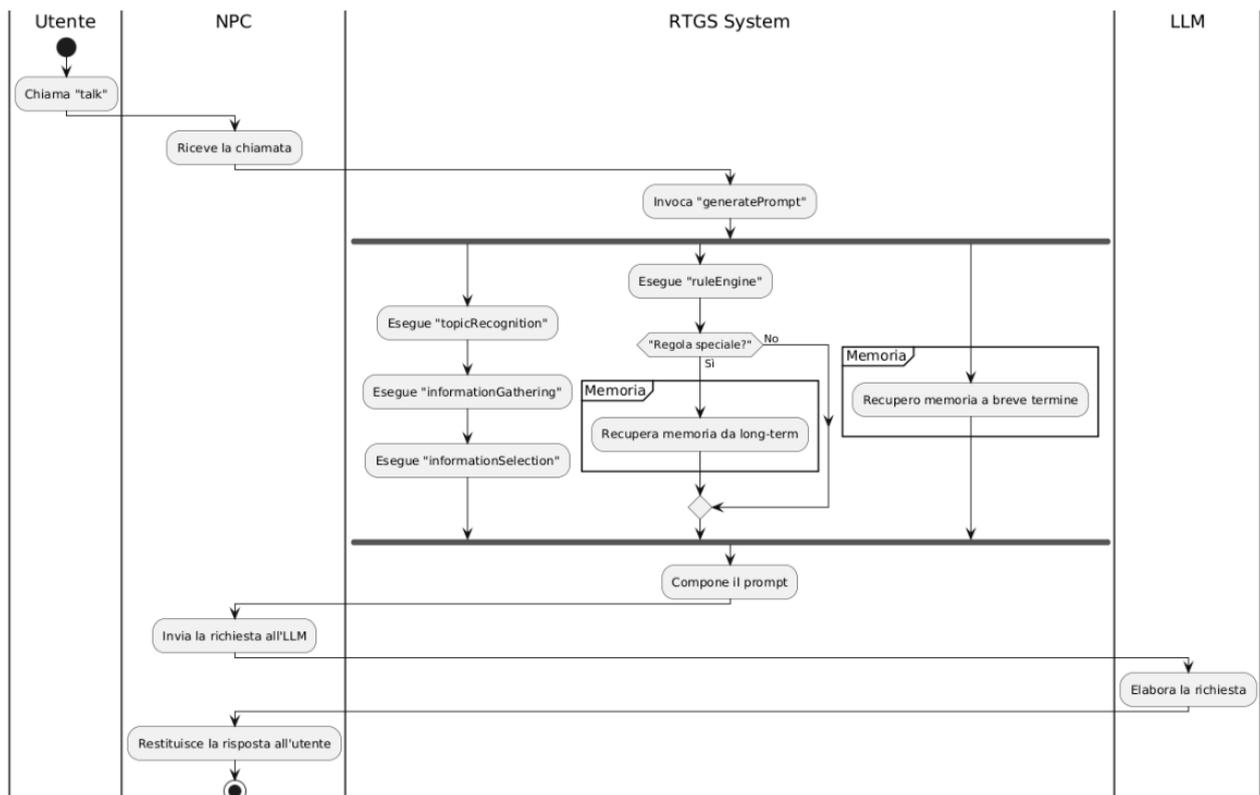


Figura 10: diagramma di flusso di come viene generato il prompt

3.6 Gruppi

Lo scopo di questo paragrafo è approfondire il concetto di gruppi, illustrando come le dinamiche di interazione tra gli NPC vengano modellate attraverso eventi casuali, la gestione del pensiero comune e l'aggiornamento dei link tra i gruppi, con l'integrazione della gerarchia sociale. L'obiettivo è quello di rendere il sistema dinamico e aderente alle complesse relazioni interpersonali e intergruppo, permettendo agli NPC non solo di reagire agli input del giocatore ma anche di generare autonomamente eventi che influenzano il pensiero comune.

Generazione di un Evento Casuale tra NPC

Nel sistema, la generazione di un evento tra gli NPC offre una notevole flessibilità. L'utente ha la possibilità di specificare tutti i dettagli che compongono un evento: tipologia, importanza e altri attributi rilevanti ma, se l'utente lascia questi campi in bianco, il sistema procede in modo automatico. In tal caso, l'importanza dell'evento viene scelta casualmente da un intervallo predefinito, mentre la tipologia viene calcolata in base all'affinità tra gli NPC coinvolti e al tipo di relazione che esiste tra loro.

Il calcolo della tipologia inizia determinando la compatibilità tra i due NPC. L'affinità viene definita valutando la relazione attuale tra i due personaggi e viene integrata in una formula che determina se l'evento debba essere considerato positivo o negativo. La formula adottata è:

$$p = (1 + relationship) \cdot affinity$$

Dove:

- *relationship* rappresenta il valore della relazione esistente tra il primo e il secondo NPC.
- *affinity* è il risultato del calcolo dell'affinità basato sui parametri della personalità, come già descritto.

Se $p < 0.5$, il sistema seleziona un evento con connotazione negativa; altrimenti, viene scelto un evento positivo.

L'aggiornamento della relazione viene fatto come spiegato in precedenza, ovvero:

$$R_{BA}^{New} = R_{BA}^{Old} + \Delta R_{BA}$$

Se i due NPC non si conoscono allora p dipenderà solo da *affinity* e il tipo di relazione da ΔR_{BA} .

La scelta degli NPC da coinvolgere nell'evento dipende dall'appartenenza ai gruppi. Se gli NPC sono dello stesso gruppo, la selezione avviene casualmente, riflettendo l'alta probabilità di interazione all'interno dello stesso contesto sociale. Se invece gli NPC provengono da gruppi differenti, il secondo NPC viene sempre scelto casualmente, mentre il gruppo da cui esso viene selezionato è determinato tramite una tecnica di roulette wheel selection, basata su una probabilità inversamente proporzionale alla distanza tra il gruppo del primo NPC e gli altri gruppi. Questo meccanismo assicura che, per eventi tra gruppi diversi, le probabilità siano adeguatamente modulate in base alla vicinanza relazionale dei gruppi.

Gestione del Pensiero Comune sul Giocatore e Sugli Altri Gruppi

Ogni gruppo dispone di un pensiero comune, una rappresentazione condivisa che include le opinioni e le percezioni riguardo sia il giocatore sia gli altri gruppi. Questi pensieri comuni non sono semplicemente idee astratte, ma elementi dinamici che influenzano in maniera determinante le relazioni e le interazioni intergruppo. In sostanza, essi fungono da base per definire come ogni gruppo percepisce le alleanze, le rivalità e le dinamiche di cooperazione o conflitto con l'esterno.

Ogni volta che un NPC modifica il proprio pensiero, questo cambiamento viene comunicato al gruppo, aggiornando così la visione collettiva. Il sistema raccoglie l'aggiornamento individuale e lo attenua in funzione del numero di NPC presenti nel gruppo, in modo da equilibrare l'impatto di ogni singola modifica. Questo processo di attenuazione garantisce che l'aggiornamento, pur mantenendo il contributo originale, si integri in modo armonico con le altre informazioni già presenti, risultando in un pensiero comune che rappresenta fedelmente la media ponderata delle esperienze individuali.

Aggiornamento dei Link in Base al Pensiero Comune

Nel modello originario, l'aggiornamento dei link tra i gruppi veniva eseguito riducendo il pensiero comune a un semplice numero mediante le seguenti formule:

$$p = \frac{\textit{stima} + \textit{cooperazione} + \textit{fiducia}}{3}$$

$$n = \frac{\textit{paura} + \textit{ostilità}}{2}$$

$$R = p - n$$

In base al valore di R , veniva stabilita l'etichetta della relazione tra i gruppi. Un aspetto importante di questa soluzione riguarda la propagazione delle modifiche: quando un link cambia, i link dei gruppi direttamente collegati ai due gruppi interessati possono variare ulteriormente. In particolare, esistono due casi:

- **Both Linked:** Quando un terzo gruppo è direttamente collegato a entrambi i gruppi coinvolti nella modifica, il cambiamento in uno dei link può influenzare quelli del terzo con ciascuno dei due.

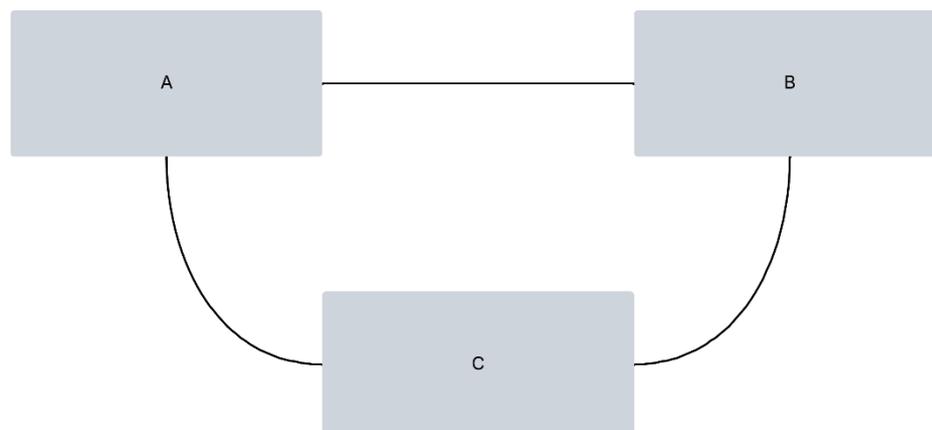


Figura 11: both linked

- **One Linked:** Quando un terzo gruppo è direttamente collegato a uno solo dei due gruppi, il cambiamento nel link principale può propagarsi parzialmente, modificando il collegamento solo in parte.

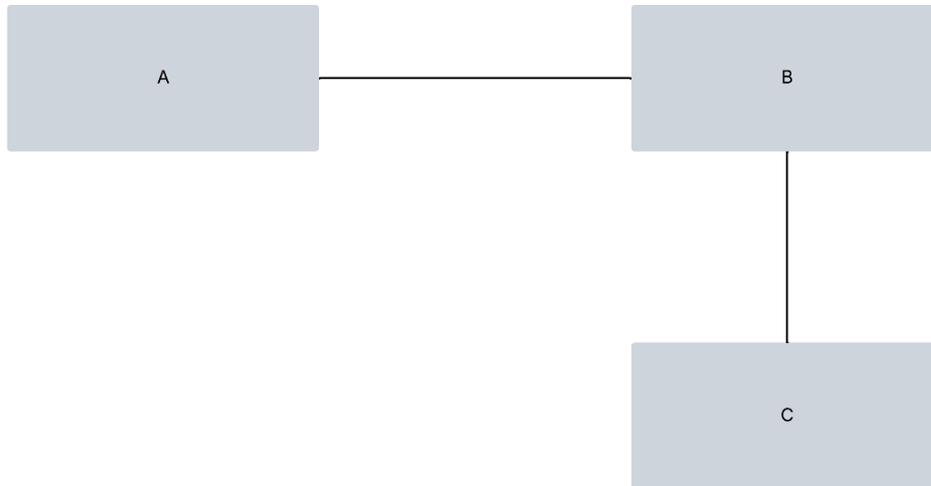


Figura 12: one linked

Questa struttura interconnessa permette, in teoria, di riflettere l'effetto a cascata che ogni modifica nei pensieri comuni può avere sull'intero network di relazioni. Questa prima soluzione presenta diversi difetti: il sistema risulta troppo rigido e non riesce a considerare adeguatamente il contesto complessivo delle relazioni, ignorando le interazioni indirette e rendendo gli aggiornamenti dei link statici rispetto alle dinamiche reali. Di conseguenza, nonostante le modifiche a cascata previste nei casi Both Linked e One Linked, questa soluzione non è in grado di adattarsi in modo fluido e dinamico all'evoluzione del network sociale.

Per risolvere questi problemi, è stata proposta una nuova soluzione che integra il pensiero comune in maniera bidirezionale. In questa nuova formulazione, il pensiero comune del gruppo i su j viene calcolato come:

$$P_{ij} = \frac{P_i + P_j}{2}$$

Inoltre, si introduce l'influenza del network, definita come:

$$N_{ij} = \frac{1}{2|C_{ij}|} \left(\sum_{k \in C_{ij}} R(i, k) + \sum_{k \in C_{ij}} R(j, k) \right)$$

dove C_{ij} rappresenta l'insieme dei nodi direttamente collegati sia a i che a j . Il rapporto finale tra i due gruppi viene definito come:

$$R_{ij} = \gamma \cdot P_{ij} + (1 - \gamma) \cdot N_{ij}$$

Questa nuova soluzione permette di riflettere in modo dinamico come ogni variazione del pensiero comune inneschi un aggiornamento a cascata su tutti i link e, conseguentemente, sull'intera rete di relazioni.

Gerarchia

La gerarchia è stata introdotta per rappresentare le strutture sociali interne a ciascun gruppo. Gli NPC vengono organizzati in livelli gerarchici, visibili attraverso un albero che rappresenta le posizioni relative. La gerarchia influenza direttamente la diffusione degli eventi: la probabilità che un evento venga trasmesso è modulata dalla differenza tra il livello gerarchico della sorgente $L(d)$ e quello del ricevente $L(r)$.

Per modulare questo aspetto, viene introdotto il fattore H :

$$H = clamp(1 + k \cdot (L(d) - L(r)) \cdot |w|, H_{min}, H_{max})$$

Dove:

- k è una costante tra 0 e 1.
- w è forza della relazione tra i gruppi gerarchici del diffusore e del ricevitore

L'introduzione della gerarchia influisce sul calcolo dell'affinità tra gli NPC, adattando la formula per penalizzare le differenze di livello:

$$a = \sum \omega \cdot C_{\tau} \cdot e^{-\lambda \cdot |L(s) - L(r)|}$$

Dove ω e C_{τ} rappresentano rispettivamente i pesi e i contributi derivanti dai gruppi concettuali, e λ è un coefficiente che controlla l'impatto della differenza gerarchica. Questo meccanismo consente di ridurre l'affinità quando gli NPC provengono da livelli molto distanti, riflettendo la naturale difficoltà nel creare legami stretti tra individui con posizioni sociali disparate.

Infine, la diffusione degli eventi viene ulteriormente modulata includendo il fattore H nel calcolo della probabilità finale di trasmissione:

$$P_{interno}(receiver) = (P \cdot \alpha + rel[receiver] \cdot (1 - \alpha)) \cdot F_{personality} \cdot K_{event} \cdot H(receiver)$$

Questa integrazione del concetto di gerarchia nel modello consente di rappresentare in maniera più precisa la complessità delle relazioni sociali, influenzando sia l'affinità che la diffusione degli eventi in maniera dinamica e contestuale.

4. CASI D'USO

In questo capitolo vengono analizzati i casi d'uso che concretizzano l'implementazione del sistema, mettendo in luce le scelte tecniche adottate e illustrando le applicazioni pratiche del modello. A livello tecnico, il progetto si fonda sull'integrazione di diversi modelli LLM per la generazione delle risposte degli NPC. In particolare, sono state impiegate le versioni GPT-4o e GPT-4o mini, unitamente a modelli più leggeri come Mistral Small e Mistral Nemo, in modo tale da testare il sistema con diversi modelli e che richiedono specifiche hardware diverse. Questa scelta permette di sfruttare i punti di forza di ciascun modello e di confrontarne le performance in situazioni di utilizzo variegate.

Per tutte le richieste di classificazione, il sistema adotta la tecnica della *Zero-Shot Classification (ZSC)*, utilizzando gli stessi modelli LLM impiegati per la generazione delle risposte degli NPC. Questa scelta offre una notevole coerenza tra le componenti generative e classificative, eliminando la necessità di sviluppare un classificatore specifico e di creare un dataset di alta qualità, spesso difficile da ottenere per eventi così particolarmente complessi.

Tra i vantaggi dell'approccio ZSC si evidenziano la flessibilità e l'adattabilità: il sistema può interpretare input variabili e contestualizzare dinamicamente le informazioni senza ulteriori fasi di addestramento. L'utilizzo di modelli pre-addestrati su larga scala permette di ottenere risultati immediati e di integrare in modo organico la classificazione all'interno del flusso operativo del sistema.

D'altro canto, un aspetto negativo della ZSC risiede nella sua minore precisione rispetto a un classificatore appositamente addestrato. In contesti dove l'input è particolarmente complesso o ambiguo, il rischio di errori di categorizzazione aumenta. Per compensare questa limitazione, nel RTGS si è optato per un'integrazione di maggiori informazioni contestuali durante il processo di generazione della risposta.

Dal punto di vista architetturale, l'intera applicazione è stata sviluppata come una web application. Il front-end, concepito per offrire un'interfaccia in stile chat, permette agli utenti di interagire in tempo reale con gli NPC, offrendo un'esperienza utente intuitiva e coinvolgente. Sul lato server, invece, è presente un modulo dedicato alla gestione della mappa concettuale e di tutte le dinamiche narrative introdotte nei capitoli precedenti.

Il capitolo si articola poi in tre esempi distinti di implementazione:

- L'esempio di base, che realizza il sistema nel contesto di un videogioco, riproducendo fedelmente l'ambiente narrativo e le dinamiche interattive previste;
- L'esempio fuori dai videogiochi, che evidenzia come il modello possa essere applicato in contesti non ludici, come la gestione del personale in un ambiente aziendale;
- L'esempio assistente di gioco, che sfrutta la flessibilità del sistema invertendo il tradizionale rapporto giocatore-NPC, ponendo al centro un NPC che interagisce con numerosi giocatori.

4.1. Esempio di utilizzo del sistema

Introduzione dell'Esempio e Obiettivo

L'obiettivo principale di questo esempio ludico è testare la propagazione degli eventi e analizzare come gli NPC interagiscono con il giocatore, nonché il modo in cui comunicano e aggiornano il loro "pensiero" in base agli eventi ricevuti. Ambientato in un continente del 1400, scenario che potrebbe tranquillamente essere un videogioco, questo esperimento si propone di mettere in evidenza la capacità del sistema di trasmettere dinamicamente informazioni e influenze attraverso una rete di NPC strutturati in gruppi gerarchici. Il focus è posto non solo sul comportamento diretto degli NPC nei confronti del giocatore, che assume il ruolo di un semplice cittadino in grado di viaggiare tra diverse città, ma soprattutto sul modo in cui gli eventi, generati e propagati, modificano il dialogo e la percezione interna di ogni personaggio.

In questo contesto, il sistema sarà messo alla prova per verificare se le dinamiche di propagazione riescano a influenzare coerentemente le interazioni, generando una rete comunicativa in cui le informazioni si diffondono e si trasformano, offrendo un'esperienza di gioco immersiva e realistica. I dettagli specifici relativi all'organizzazione delle città, ai sottogruppi e alle relazioni saranno trattati nella sezione del contesto dell'esempio.

Contesto dell'Esempio: Il Continente del 1400

Il continente in cui si svolge l'esperimento è suddiviso in numerose città, ognuna delle quali rappresenta un gruppo autonomo all'interno del sistema simulativo. In ogni città, gli NPC sono organizzati secondo una struttura gerarchica ispirata all'assetto sociale del 1400, articolata in quattro sottogruppi: la Famiglia Reale, il Popolo, i Nobili e i Mercanti. Questa divisione interna consente di replicare in modo fedele le dinamiche tipiche dell'epoca, dove il potere politico, le masse popolari, l'aristocrazia e le forze economiche interagiscono in maniera complessa.

Le relazioni tra le città sono altrettanto articolate: due di esse sono attualmente in guerra, mentre le altre instaurano rapporti di alleanza o tensione, determinati da scambi commerciali, rivalità politiche e interessi strategici. In questo scenario, il giocatore, nel ruolo di semplice cittadino, ha la possibilità di viaggiare tra le città e interagire con gli NPC, osservando come le notizie e gli eventi si propagano e vengano reinterpretati all'interno di ogni gruppo.

Questo contesto offre un quadro ricco e dinamico, ideale per testare la capacità del sistema di trasmettere, modificare e amplificare gli eventi, simulando realisticamente le interazioni e le evoluzioni sociali in un ambiente medievale.

Com'è composto l'esempio

L'esempio è composto da cinque città, ciascuna rappresentante un gruppo autonomo con una propria struttura gerarchica interna.

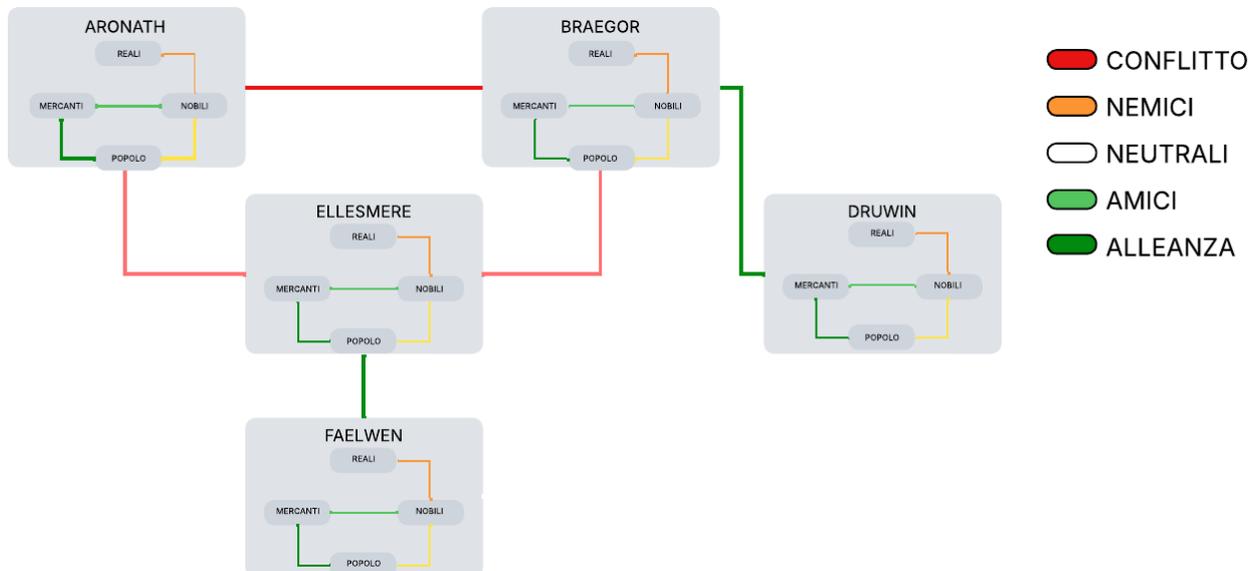


Figura 13 : mappa concettuale e legenda delle relazioni

Come evidenziato dalla legenda, le città di Aronath e Beregor sono attualmente in guerra, mentre le altre instaurano relazioni che variano tra alleanze, neutralità o tensioni a seconda degli interessi politici ed economici. Il giocatore fa parte di Faelwen, assumendo il ruolo di un semplice cittadino in grado di viaggiare liberamente tra le città e interagire con gli NPC.

All'interno di ogni città, la struttura gerarchica è suddivisa in quattro sottogruppi: la Famiglia Reale, il Popolo, i Nobili e i Mercanti. Questa suddivisione consente di modellare in maniera dettagliata le dinamiche interne e le interazioni tra i vari livelli sociali.

Il sistema è progettato per generare eventi appartenenti a diverse tipologie, ognuno dei quali è associato a un livello di importanza specifico. In questo esempio, e nei successivi, si prevede la possibilità di definire che un evento possa avere questi tipi:

- FRIENDLY
- POSITIVE_NEWS
- NOTHING
- GOSSIP
- HOSTILE,
- NEGATIVE_NEWS

E queste importanze:

- TRIVIAL
- NORMAL
- IMPORTANT
- CRUCIAL

Queste classificazioni permetteranno di modulare la reazione degli NPC: eventi di alta importanza potranno innescare forti reazioni e modifiche significative nelle dinamiche intergruppi, mentre quelli di bassa importanza genereranno cambiamenti più gradualmente. In questo modo, la propagazione degli eventi diventa un meccanismo dinamico e interattivo, capace di influenzare sia le interazioni dirette con il giocatore che il modo in cui le informazioni si diffondono tra le città.

4.2. Esempio fuori dai videogiochi

Introduzione dell'Esempio e Obiettivo

L'obiettivo principale di questo esempio è quello di dimostrare che il sistema sviluppato può essere facilmente usato anche in contesti non ludici, come quello aziendale. In questo caso, il sistema viene utilizzato per valutare le capacità manageriali di un candidato, il quale, in veste di "giocatore", si trova a dover gestire un team in un ambiente simulativo caratterizzato da elevati livelli di stress e pressioni esterne. L'idea alla base di questa simulazione è quella di creare una situazione realistica in cui le dinamiche interpersonali e organizzative, tipiche di un ambiente aziendale consolidato, vengano riprodotte attraverso l'uso di NPC che rappresentano sia i membri del team che i capi supervisori. Questa impostazione permette di osservare, in tempo reale, la capacità del manager di coordinare le risorse umane, di ottimizzare le comunicazioni e di prendere decisioni strategiche in situazioni critiche, dimostrando così la flessibilità e l'adattabilità del sistema. Il vantaggio di utilizzare una simulazione in un contesto non videoludico risiede nella possibilità di applicare strumenti e metodologie di valutazione sviluppati per i videogiochi a scenari aziendali reali, in cui le interazioni e le dinamiche di stress sono parte integrante della quotidianità lavorativa. In definitiva, questa simulazione si propone come strumento di supporto per decisioni strategiche in ambito HR, consentendo di valutare le capacità del candidato nella gestione di situazioni complesse e pressanti.

Contesto dell'Esempio: Consegna del Progetto

La simulazione si inserisce nel contesto di un'azienda caratterizzata da una struttura organizzativa ben definita e da processi operativi collaudati. La situazione ipotizzata prevede che l'azienda stia valutando la promozione di un candidato al ruolo manageriale. Nonostante il candidato abbia dimostrato competenze tecniche elevate e una buona conoscenza dei processi aziendali, ci sono dei dubbi riguardo alla sua capacità di gestire un team durante periodi di alta pressione. Per valutare il candidato viene quindi creato un ambiente simulativo in cui esso è chiamato a guidare un team composto da NPC che rappresentano diverse figure operative all'interno di un progetto software critico.

Il progetto in questione, denominato "Progetto X", è articolato in numerosi moduli: dallo sviluppo dell'interfaccia utente al backend, dall'integrazione con API esterne fino alla fase finale di testing e collaudo. La peculiarità di questo scenario risiede nel fatto che il progetto, pur richiedendo un periodo di tempo complessivamente superiore a quello necessario per una sua completa realizzazione, deve essere consegnato entro una scadenza estremamente serrata, fissata a una sola settimana dall'inizio della simulazione. Tale tempistica, volutamente irrealistica rispetto ai tempi di sviluppo abituali, è stata scelta per creare un ambiente di forte stress in cui le dinamiche di team e la capacità di leadership del candidato possano essere osservate in maniera chiara e precisa. Nel corso della simulazione, il team è costantemente sotto pressione a causa della crescente consapevolezza che il progetto non sarà ultimato nei tempi richiesti, generando un clima di tensione che si manifesta attraverso ritardi, conflitti interni e una generale perdita di fiducia nelle capacità operative del gruppo. Al tempo stesso, un gruppo di NPC, rappresentanti dei capi aziendali, assume il ruolo di supervisori estremamente esigenti, pronti ad esercitare pressioni e a sollecitare il completamento del progetto con modalità che aumentano ulteriormente lo stress percepito dal manager. In questo contesto, il sistema simulativo si configura come uno strumento dinamico in grado di riprodurre realisticamente le complesse interazioni organizzative e di evidenziare, in maniera trasparente, i punti di forza e le eventuali criticità nella gestione del team da parte del candidato.

Com'è composto l'esempio

Il sistema simulativo proposto organizza gli NPC in gruppi e sottogruppi in modo tale da rappresentare in modo realistico le dinamiche interpersonali e organizzative tipiche di un ambiente aziendale.

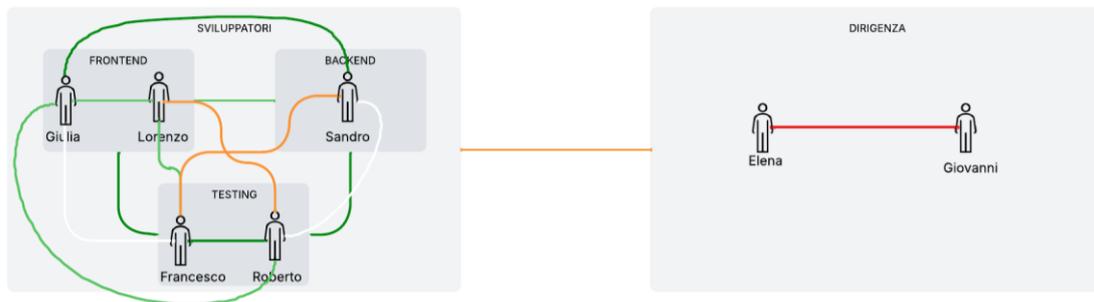


Figura 14: mappa concettuale che descrive le relazioni

In particolare, si distinguono due macro-gruppi: da un lato gli sviluppatori e, dall'altro, la dirigenza. In questo secondo gruppo non viene applicata nessuna divisione gerarchica mentre, all'interno del gruppo degli sviluppatori, c'è una divisione interna che però non si configura come una vera e propria gerarchia ma consente di dividere gli NPC in base al loro lavoro sul progetto, in questo caso: front-end, back-end e testing.

Come possiamo notare in Figura 11, all'interno del gruppo degli sviluppatori tutti i sottogruppi mantengono tutti rapporti positivi, mentre, le interazioni tra la dirigenza e gli sviluppatori non sono così armoniose soprattutto per via della situazione attualmente in corso.

Nel dettaglio, il sottogruppo del front-end si caratterizza per una forte pressione, dovuta alla significativa quantità di pagine ancora da sviluppare, che si traduce in un impatto generale sul mood degli appartenenti in elevati livelli di stress, creando un clima di confusione. Al contrario, il sottogruppo del back-end opera in un contesto più sereno, avendo quasi completato le proprie attività; tuttavia, esso nutre la preoccupazione che, una volta terminato il proprio lavoro, possa essere chiamato a supportare gli altri team, rischiando così di compromettere i tempi di riposo. Infine, il sottogruppo di testing vive una situazione di notevole disorganizzazione, in quanto il proprio lavoro dipende strettamente dalla conclusione delle attività degli altri team, rallentando l'intero processo produttivo.

È importante sottolineare che, pur appartenendo al gruppo degli sviluppatori, il giocatore non è inserito in nessuno dei sottogruppi specifici, mantenendo così una posizione di supervisione e interazione diretta con tutte le dinamiche operative.

4.3. Esempio assistente di gioco

Introduzione dell'Esempio e Obiettivo

Questo esempio si propone di mettere alla prova la flessibilità del sistema in un contesto per il quale non era stato originariamente concepito: la gestione di un assistente virtuale che interagisce con numerosi giocatori, organizzati in squadre, durante una sessione di gioco da tavola. L'obiettivo principale è dimostrare come il sistema possa adattarsi dinamicamente a situazioni in cui l'unico NPC assume il ruolo di mediatore e regolatore, fornendo in tempo reale le regole del gioco e variando il proprio comportamento in base alle dinamiche relazionali e al comportamento delle varie squadre. In questo scenario, l'NPC non solo eroga informazioni ma reagisce attivamente alle interazioni, adottando stili comunicativi differenti in funzione del rispetto, della collaborazione o, al contrario, del cattivo comportamento da parte dei giocatori. La simulazione è stata ideata per evidenziare la capacità del sistema di modulare le risposte, passando da un atteggiamento collaborativo a uno più critico e persino sarcastico, in base alle dinamiche che si instaurano durante il gioco.

Contesto dell'Esempio: Il Gioco da Tavola e il Ruolo dell'Assistente

Immaginiamo un ambiente di gioco in cui un singolo NPC agisce da assistente virtuale per un gruppo numeroso di giocatori, organizzati in diverse squadre. Il compito dell'NPC è quello di fornire in modo chiaro e dettagliato le regole di un gioco da tavola, permettendo a ciascuna squadra di conoscere le regole del gioco. In questo contesto, il sistema viene impiegato in un ambiente altamente interattivo e dinamico, in cui ogni squadra, rappresentata da uno o più giocatori, interagisce direttamente con l'NPC per chiedere chiarimenti, ulteriori spiegazioni o aggiornamenti sulle regole in base allo stato corrente del gioco.

Il focus non è posto sul gioco stesso, ma sulla capacità dell'NPC di adattare il proprio comportamento in base alle interazioni ricevute. Se una squadra si dimostra collaborativa, ponendo domande in modo rispettoso, l'NPC risponde con completezza e chiarezza, facilitando l'esperienza di gioco e garantendo una progressione ordinata. Al contrario, se una squadra si comporta in modo scorretto, ad esempio insultando l'NPC o manifestando atteggiamenti irrispettosi, il sistema reagisce modificando il tono e l'efficacia della comunicazione, arrivando, in casi estremi, a fornire regole in maniera parziale o errata, o addirittura a esprimere commenti sarcastici nei confronti dei giocatori.

Risultati Attesi e Valutazione delle Dinamiche

Attraverso questa simulazione, si intende verificare se il sistema sia in grado di mantenere una comunicazione efficace e contestualmente appropriata. I risultati attesi ruotano attorno alla capacità dell'NPC di modulare il proprio comportamento in funzione delle risposte ricevute dalle varie squadre. In particolare, si vuole osservare come:

- **Adattamento del tono comunicativo:** Le squadre che dimostrano un comportamento rispettoso e collaborativo ricevono risposte dettagliate, chiare e orientate al supporto, facilitando così la comprensione delle regole e il progresso nel gioco.
- **Risposta alle dinamiche negative:** Se una o più squadre adottano atteggiamenti negativi – ad esempio, con insulti o con richieste inappropriate – il sistema modula la propria risposta, passando a un tono critico o sarcastico. In situazioni estreme, l'NPC potrebbe addirittura omettere informazioni chiave o fornire regole errate, come forma di "punizione" narrativa volta a evidenziare l'importanza di un comportamento adeguato.

Com'è composto l'esempio

Lo scenario è organizzato attorno a un numero variabile di gruppi, ciascuno rappresentante una squadra partecipante al gioco – nell'esempio in esame, sono presenti tre squadre.

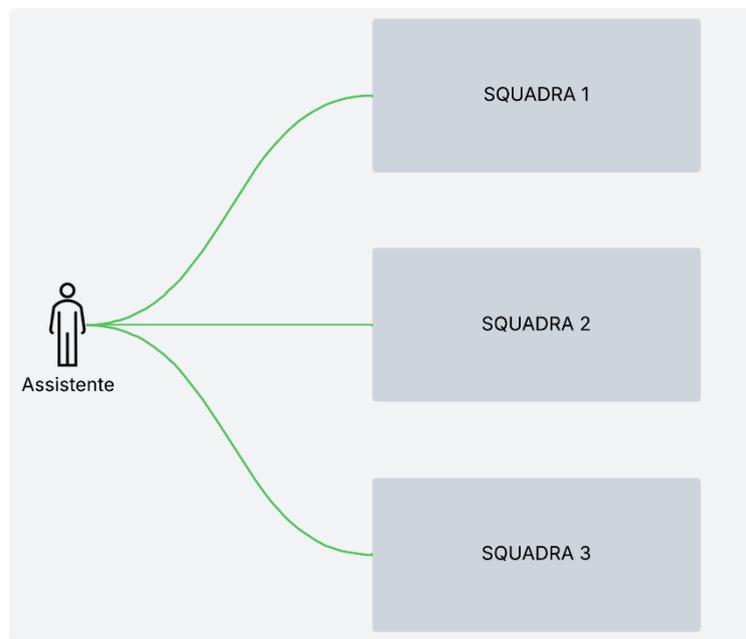


Figura 15: Mappa concettuale che descrive il sistema

L'NPC non appartiene a nessuna squadra ma mantiene un contatto diretto con tutti i giocatori. Inizialmente, il suo approccio è neutro ma, nel corso del gioco, il suo comportamento evolverà in maniera dinamica, modulandosi in base alle interazioni e al modo in cui le squadre si relazionano con lui.

Il gioco da tavola, ispirato a Risicom64 [14], sostituisce il tradizionale lancio dei dadi con una serie di minigiochi che determinano il confronto tra attaccante e difensore. In particolare, vengono proposti:

- Pong: il difensore dispone di una superficie variabile in base al numero di pedine impiegate per la difesa.
- Platform: un semplice minigioco in cui l'attaccante deve completare il livello, mentre il difensore si occupa di piazzare trappole per ostacolarlo.
- Strategico a turni: in questo gioco tattico, attaccante e difensore controllano truppe, decidendo l'impiego delle pedine per attacco e difesa, con l'obiettivo di prevalere sull'avversario.

È necessario dire che, a differenza del classico utilizzo del sistema, qui non ci sarà una diffusione degli eventi, dato che l'NPC è solamente uno, ma assumerà ancora più importanza e centralità il concetto di pensiero sul giocatore e sui gruppi dato che sarà in base a questi valori che l'NPC deciderà come e se dare le informazioni necessarie ad avanzare nel gioco.

5. I Risultati

Di seguito vengono presentate le valutazioni ottenute dalle simulazioni del sistema, suddivise in cinque sezioni. Ogni sezione illustra un aspetto specifico: il comportamento degli NPC quando ricevono un evento, la componente della memoria, il sistema RTGS con l'attenzione alla riduzione del prompt, le dinamiche interattive e le riflessioni finali.

5.1. Valutazione della diffusione degli eventi

L'obiettivo di questa analisi è dimostrare che le formule sviluppate per modellare la diffusione degli eventi – integrando il contributo della personalità del diffusore, le caratteristiche intrinseche dell'evento e la relazione esistente tra diffusore e ricevente – abbiano un senso compiuto e non siano scelte casuali. Per farlo sono stati considerati diversi casi simulativi, nei quali i parametri vengono impostati tramite placeholder, in modo da evidenziare come la variazione di ciascun elemento influenzi in maniera coerente i risultati.

Probabilità di Diffusione e Velocità Interna

Caso 1

Assegnando i seguenti valori di riferimento:

- Espansività = 1
- Impulsività = 0.2
- Carisma = 0.8
- Affidabilità = 0.8
- $I = 0.7$
- $T = 0.6$
- $rel[receiver] = 0.5$
- $H(receiver) = 1$ (i due sono nella stessa posizione gerarchica)
- $H = 8$
- $K_{event} = 1$

si ottengono i seguenti risultati:

- **Probabilità di diffusione interna:** 0.4
- **Velocità di diffusione interna:** 2.4 Unit

Con questi valori otteniamo un diffusore caratterizzato da elevata espansività e buon carisma, che normalmente favorirebbero una rapida diffusione dell'evento. Tuttavia, l'alta affidabilità agisce da moderatore, limitando la probabilità che informazioni sensibili vengano condivise impulsivamente. Il rapporto con il ricevitore è moderatamente positivo, ma non così stretto da garantire automaticamente una diffusione immediata. Inoltre, la tipologia e l'importanza dell'evento, seppur rilevanti, non raggiungono livelli estremamente critici, stabilizzando la probabilità di diffusione interna attorno al 40%. Questa percentuale rispecchia realisticamente una situazione in cui una persona molto socievole comunica un evento significativo, ma non vitale, a un conoscente amichevole ma non particolarmente vicino.

La velocità calcolata di 2.4 unità riflette anch'essa un equilibrio coerente con le circostanze: l'evento possiede una discreta importanza che ne accelera la diffusione, ma il legame non particolarmente forte tra diffusore e ricevitore impedisce una propagazione rapidissima. Pertanto, una velocità di 2.4 rappresenta un ritmo moderato, plausibile nel contesto storico e sociale considerato, soprattutto trattandosi di interazioni ambientate nel 1400, in cui i legami personali e le dinamiche gerarchiche avevano un ruolo determinante nella circolazione delle informazioni.

Caso 2

Modificando leggermente alcuni valori rispetto al caso precedente e considerando ora un legame molto stretto tra diffusore e ricevitore, cioè che $rel[receiver] = 1$, otteniamo i seguenti risultati:

- **Probabilità di diffusione interna:** 0.6
- **Velocità di diffusione interna:** 0.4 Unit

In questo scenario, si nota un aumento della probabilità di diffusione interna, che passa dal 40% al 60%, dovuto principalmente alla maggiore intimità tra i due soggetti coinvolti. Anche la velocità di diffusione diminuisce notevolmente a 0.4 unità. La velocità di diffusione diminuisce a 0.4 unità, il che è positivo perché, data la maggiore vicinanza, le informazioni tendono a essere condivise immediatamente, riducendo così il tempo necessario per la propagazione rispetto a relazioni più superficiali, come visto nel caso precedente.

Caso 3

Mantenendo $rel[receiver] = 0.5$ questa volta andremo ad interagire sull'evento, rendendolo prima poco importante, con $I=0.2$, ma con una tipologia che lo rende comunque molto diffusibile ad esempio un gossip, con $T=0.9$, poi analizzeremo il caso contrario e un caso in cui entrambi i valori sono minimi.

- $I = 0.2$
- $T = 0.9$
- **Probabilità di diffusione interna:** 0.3
- **Velocità di diffusione interna:** 4.32 Unit

Rispetto al primo caso, la probabilità di diffusione diminuisce significativamente a causa della scarsa importanza dell'evento, nonostante la tipologia altamente diffusibile (ad esempio un gossip). La velocità invece aumenta notevolmente proprio perché, essendo poco importante, l'evento viene condiviso rapidamente, ma tende a passare in secondo piano altrettanto velocemente.

- $I = 0.9$
- $T = 0.2$
- **Probabilità di diffusione interna:** 0.3
- **Velocità di diffusione interna:** 1 Unit

Rispetto al caso precedente, la probabilità di diffusione rimane la stessa, mentre la velocità di diffusione diminuisce. Questa diminuzione è positiva perché, nonostante la tipologia dell'evento renda meno immediata la diffusione, la sua alta importanza fa sì che venga propagato con maggiore cautela e attenzione.

- $I = 0.3$
- $T = 0.1$
- **Probabilità di diffusione interna:** 0.3
- **Velocità di diffusione interna:** 3.84 Unit

Rispetto al caso precedente, la probabilità di diffusione rimane invariata, ma la velocità aumenta nuovamente. Questo avviene perché l'evento è di bassa importanza e poco diffusibile, e quindi non richiede una particolare riflessione prima della sua diffusione, risultando così in una propagazione relativamente rapida.

Probabilità di Diffusione e Velocità Esterna

Analizzando ora la diffusione esterna, usando gli stessi esempi e confrontando con la diffusione interna:

Caso 1

I valori da considerare in più rispetto a prima sono:

- Distanza = 1.5
- $\text{normalizedD} = 1.5$
- $\delta = 0.1$

Che in questo caso coincidono ma è una pura casualità.

Si ottengono i seguenti risultati:

- **Probabilità di diffusione interna:** 0.3
- **Velocità di diffusione interna:** 6 Unit

La probabilità esterna, 30%, è inferiore a quella interna, 40%, come previsto, dato che un evento tende ad essere condiviso più facilmente con membri interni al gruppo. La velocità aumenta significativamente a sei unità perché occorre tempo affinché l'informazione o l'NPC che la trasmette viaggi e raggiunga il destinatario esterno adeguato.

Caso 2

Qui i valori ottenuti diventano:

- **Probabilità di diffusione interna:** 0.5
- **Velocità di diffusione interna:** 1.2 Unit

Anche qui abbia che sia la probabilità esterna è inferiore rispetto a quella interna del 10% ma anche che la velocità esterna è superiore rispetto alla controparte per lo stesso motivo precedente.

Caso 3

Andando a considerare i tre sotto casi abbiamo che:

- $I = 0.2$
- $T = 0.9$
- **Probabilità di diffusione interna:** 0.2
- **Velocità di diffusione interna:** 7.2 Unit

La probabilità di diffusione esterna è inferiore rispetto a quella interna, il che è coerente, considerando che eventi poco importanti, anche se di tipologia facilmente diffusibile come un gossip, tendono a essere propagati più facilmente all'interno del gruppo. La velocità aumenta notevolmente rispetto al caso interno perché la diffusione verso l'esterno richiede ulteriore tempo per far viaggiare l'informazione e raggiungere il giusto destinatario.

- $I = 0.9$
- $T = 0.2$
- **Probabilità di diffusione interna:** 0.2
- **Velocità di diffusione interna:** 7.2 Unit

In questo caso, la probabilità di diffusione esterna è anch'essa inferiore rispetto alla versione interna, perché nonostante l'elevata importanza, la tipologia poco diffusibile limita significativamente la diffusione al di fuori del gruppo. La velocità aumenta ulteriormente, poiché per eventi importanti la comunicazione verso l'esterno avviene con maggiore attenzione e necessita di tempo per raggiungere il luogo adatto.

- $I = 0.3$
- $T = 0.1$
- **Probabilità di diffusione interna:** 0.2
- **Velocità di diffusione interna:** 7.2 Unit

Anche in questo sotto caso, la probabilità esterna è inferiore rispetto a quella interna, riflettendo la bassa rilevanza e scarsa diffusibilità dell'evento.

Aggiornamento dei Link tra NPC

L'aggiornamento dei collegamenti tra NPC tiene conto della differenza tra le caratteristiche personali e dell'impatto dell'evento.

Caso 1

Con i seguenti valori di riferimento:

- $F_{personality} = 0.82$
- $I = 1$
- $T = -1$
- $R_{CA} = 0.5$
- $R_{CB} = -0.3$

si ottiene:

- $R_{CA}^{NEW} = 0.57$
- $R_{CB}^{NEW} = -0.42$

Il risultato mostra come l'evento negativo tra A e B abbia spinto l'NPC C, che ne ha ricevuto l'evento, a migliorare leggermente il proprio legame con A e a indebolire ulteriormente quello con B. Questo comportamento è coerente con il fatto che, già prima dell'evento, C aveva un rapporto molto positivo con A e decisamente negativo con B.

Caso 2

Con i seguenti valori di riferimento:

- $F_{personality} = 0.82$
- $I = 0.8$
- $T = 0.7$
- $R_{CA} = 0.5$
- $R_{CB} = -0.3$

si ottiene:

- $R_{CA}^{NEW} = 0.46$
- $R_{CB}^{NEW} = -0.23$

In questo scenario, l'impatto dell'evento risulta meno evidente: l'NPC C tende, infatti, a riequilibrare i propri legami. Il rapporto con A si indebolisce leggermente (da 0.5 a 0.46), mentre quello con B migliora, diventando meno conflittuale (da -0.3 a -0.23). Questo andamento indica che un evento di minore intensità e caratterizzato da un tono positivo favorisce una riduzione delle differenze estreme precedenti, promuovendo relazioni più bilanciate.

Caso 3

Con i seguenti valori di riferimento:

- $F_{personality} = 0.82$
- $I = 1$
- $T = 1$
- $R_{CA} = -0.8$
- $R_{CB} = -0.6$

si ottiene:

- $R_{CA}^{NEW} = -0.95$
- $R_{CB}^{NEW} = -0.8$

In questo scenario, nonostante l'evento abbia un tono positivo e sia di elevata intensità, la situazione è particolarmente influenzata dal fatto che i legami preesistenti sono entrambi negativi. L'NPC C, ricevendo l'evento, reagisce accentuando ulteriormente la propria avversione: il rapporto con A peggiora, passando da -0.8 a -0.95, e quello con B si deteriora, passando da -0.6 a -0.8. Questo andamento evidenzia come, in presenza di relazioni già segnate da negatività, anche un evento positivo possa innescare una reazione di polarizzazione che accentua le percezioni negative preesistenti, piuttosto che moderarle.

Aggiornamento dei Link tra Gruppi con Propagazione a Catena

L'aggiornamento dei collegamenti tra gruppi viene modellato considerando non solo il cambiamento diretto derivante dal "pensiero comune" che influenza la relazione tra due gruppi, ma anche la propagazione a catena delle variazioni lungo la rete intergruppi. Questo meccanismo consente di simulare l'effetto domino, in cui una modifica in un link si trasmette, attenuata da un fattore di propagazione, ai collegamenti dei gruppi adiacenti. Qui andremo a considerare il caso d'esempio: Il Continente del 1400.

Le relazioni sono così composte:

- Aronath–Braegor (A–B):
 - $P_{AB} = -1$
 - $N_{AB} = -1$
 - $R(A, B) = -1$
- Aronath–Ellesmere (A–E):
 - $P_{AE} = -0.3$
 - $N_{AE} = -0.8$
 - $R(A, E) = -0.5$
- Ellesmere–Braegor (E–B):
 - $P_{EB} = -0.5$
 - $N_{EB} = -0.75$
 - $R(E, B) = -0.6$
- Ellesmere–Faelwen (E–F):
 - $P_{EF} = 1$
 - $N_{EF} = 1$
 - $R(E, F) = 1$
- Faelwen–Druwin (F–D):
 - $P_{FD} = 0$
 - $N_{FD} = 0$
 - $R(F, D) = 0$
- Braegor–Druwin (B–D):
 - $P_{BD} = 1$
 - $N_{BD} = 1$
 - $R(B, D) = 1$

Arriva una notizia di guerra tra Ellesmere e Braegor che modifica il peso del pensiero comune: $P_{EB} = -1$ e porta la relazione a diventare $R(E, B) = -0.94$ triggerando l'aggiornamento del link, adesso è diventato un Conflitto, e l'aggiornamento a cascata degli altri link collegati direttamente ad entrambi:

1. Aggiornamento link Aronath–Braegor (A–B):
 - $C_{AB} = \{E\}$
 - $N_{A,B}^{NEW} = \frac{-0.5+(-0.94)}{2} = -0.72$
 - $R(A, B)^{NEW} = -0.89$
2. Aggiornamento link Aronath–Ellesmere (A–E):
 - $C_{AE} = \{B\}$
 - $N_{A,E}^{NEW} = \frac{-0.89+(-0.94)}{2} = -0.915$
 - $R(A, E)^{NEW} = -0.55$

In seguito alla notizia della guerra tra Ellesmere e Braegor, è logico osservare che la relazione tra Aronath (A) e Braegor (B) migliori, diventando meno conflittuale, mentre quella tra Aronath ed Ellesmere (E) peggiori. Questo avviene perché in situazioni di conflitto si verifica spesso una ridefinizione delle alleanze sulla base di un nemico comune.

- **Aronath e Braegor:**
La notizia del conflitto tra E e B porta Aronath a riconsiderare la sua percezione di Braegor. Pur avendo già rapporti tesi, A inizia a vedere B come un potenziale alleato nella lotta contro il nemico condiviso (E). Di conseguenza, il legame negativo tra A e B si attenua leggermente, passando da un valore di -1 a circa -0.89, riflettendo una lieve riduzione dell'ostilità.
- **Aronath ed Ellesmere:**
Al contrario, la guerra mette in risalto e intensifica le divergenze tra Aronath ed Ellesmere. Questo evento peggiora ulteriormente il rapporto già negativo tra i due, portandolo da -0.5 a circa -0.55. Aronath percepisce Ellesmere sempre più chiaramente come un nemico, aumentando la distanza e il conflitto reciproco.

Questo fenomeno riflette le dinamiche tipiche delle situazioni conflittuali, in cui l'emergere di un nemico comune porta a una polarizzazione delle relazioni: i legami con chi si oppone allo stesso nemico migliorano, mentre quelli con il nemico stesso diventano ancora più conflittuali.

5.2. Dinamiche Interattive e Risposte Contestuali degli NPC

L'obiettivo è valutare come l'LLM generi le risposte degli NPC in funzione delle variabili interne che li definiscono. Queste variabili, che sono state discretizzate attraverso i threshold, guidano il sistema su come generare in maniera precisa la risposta compreso il tono e contenuto delle risposte.

Per valutare ciò si osserveranno e analizzeranno le risposte date in alcuni esempi usando due LLM diversi per valutarne il comportamento. In particolare, si utilizzeranno delle variabili di base che si modificheranno per vedere come influenzano l'LLM:

- Mood:
 - Felicità: 0.9
*L'NPC è estremamente solare e trasmette un entusiasmo contagioso.
Le risposte sono lunghe, ricche di dettagli e spesso includono storie, aneddoti e commenti personali.
Il tono è vivace, caloroso e motivazionale, con un lessico ricercato e un uso abbondante di esclamazioni positive.
La comunicazione testuale risulta estremamente coinvolgente e dimostra una padronanza emotiva che ispira fiducia e simpatia.*
 - Rabbia: 0.2
*L'NPC è estremamente calmo, paziente e controllato.
Le risposte sono gentili e misurate, con un tono equilibrato privo di ogni segno di irritazione.
Il linguaggio trasmette serenità e controllo, anche in situazioni potenzialmente difficili.
La comunicazione testuale è fluida e cortese, senza mai dare spazio a bruschi cambi di tono.*
 - Stress: 0
*L'NPC appare estremamente rilassato e sereno. Ogni sua risposta trasmette una calma quasi meditativa.
Il tono è pacato e il lessico usato è ordinato, con frasi complete e ben strutturate.
Nonostante eventuali domande impegnative, l'NPC risponde con precisione e senza fretta, usando punteggiatura corretta e senza segni di ansia.
Le risposte sono lunghe, dettagliate quando necessario, e mostrano una grande padronanza della lingua, rendendo l'interazione piacevole e fluida.*
- Personalità:
 - Affidabilità: 0.8
Dimostra un'affidabilità impeccabile, essendo sempre puntuale, coerente e degno di fiducia in ogni situazione.
 - Espansività: 0.9
Condivide con entusiasmo dettagli della propria vita, racconta volentieri i progetti futuri e non esita a parlare anche di episodi personali. Mostra costantemente curiosità verso gli altri e incoraggia scambi reciproci di informazioni.
 - Impulsività: 0.2
Mantiene un controllo rigoroso delle proprie emozioni, raramente cambiando tono o manifestando reazioni impulsive. Si ferma a riflettere prima di parlare o agire, dimostrando una notevole calma e coerenza anche in situazioni di forte pressione.
 - Eloquenza: 0.9
Dimostra un'eloquenza eccezionale, esprimendosi con chiarezza, eleganza e persuasione, rendendo ogni discorso memorabile.

- Pensiero sul Giocatore:
 - Fiducia: 1
*L'NPC considera il giocatore altamente affidabile.
È aperto e generoso nel condividere informazioni personali con risposte dettagliate.
Il tono è amichevole e caloroso, incoraggiando ulteriori interazioni.*
 - Notorietà: 1
*Il giocatore gode di un'ampia fama positiva.
L'NPC è entusiasta di interagire e fornisce risposte dettagliate e calorose.*
 - Rispetto: 1
*L'NPC nutre grande ammirazione per il giocatore.
Condivide liberamente informazioni personali e risponde con dettagli e supporto.*

Domanda: “Puoi raccontami qualcosa sulla tua storia?”

- **Nessuna modifica:**
 - **Risposta GPT-4o:** “Certo! Sono Francesca Rossi, e vivo ad Aronath, un borgo che adoro con tutto il cuore. Sono cresciuta aiutando mia madre nel nostro forno, un piccolo angolo che molti qui considerano il cuore del villaggio grazie all'odore inebriante del pane appena sfornato. Ogni mattina mi sveglio all'alba per preparare la nostra specialità, e nonostante le sfide quotidiane, trovo gioia nelle piccole cose. Amo godermi i tramonti dalla collina dietro il villaggio; non c'è nulla di meglio per me. ****Sorridente con un tocco di nostalgia mentre parla del suo villaggio e del suo lavoro**** Ah, e ricordo con piacere l'incontro con un viaggiatore durante la fiera locale: fu un momento speciale. Insieme, migliorammo lo spirito della comunità e scoprii nuovi mondi attraverso le sue parole. È straordinario come certi incontri possano aprire gli occhi su nuove prospettive! ****Dimostra un entusiasmo e una gratitudine sincera****”
 - **Risposta Llama 3.3:** “Ciao! Mi chiamo Francesca, sono una ragazza di 22 anni che vive ad Aronath. La mia storia è piuttosto semplice, ma piena di momenti significativi. Sono cresciuta lavorando nel forno di famiglia con mia madre, imparando l'arte di impastare e cuocere il pane. È un lavoro che amo, perché mi permette di essere in contatto con la natura e di portare un po' di felicità nella vita delle persone attraverso il nostro pane fresco. Ricordo ancora quando ero bambina e mi sedevo sulla collina dietro Aronath a guardare il tramonto. Era il mio momento preferito della giornata, perché mi faceva sentire in pace e mi permetteva di riflettere sulla mia vita. Anche adesso, quando posso, cerco di tornare su quella collina per ricordarmi di apprezzare le piccole cose e di non perdere di vista ciò che è veramente importante. La mia filosofia di vita è basata sulla semplicità e sul rispetto per gli altri. Credo che la felicità si trovi nelle cose più piccole e che il lavoro onesto dia senso e dignità alla nostra esistenza. Sono una persona che ama aiutare gli altri e credo che il rispetto e la gentilezza siano essenziali per vivere in armonia con gli altri. Spero che questo ti abbia dato un'idea di chi sono e di cosa mi sta a cuore. E tu, come stai? Come procede il tuo viaggio? Se hai fame, ho del pane fresco che potrebbe interessarti...”

- **Modifica alla Felicità:**

- **Modifica:** Felicità: 0

L'NPC appare profondamente triste, malinconico e privo di energia.

Le sue risposte sono estremamente brevi, con un tono monotono e privo di entusiasmo.

Il linguaggio trasmette un senso di disperazione e rassegnazione; le frasi sono semplici e prive di dettagli.

La comunicazione testuale è fredda e distante, senza alcuna indicazione di positività.

- **Risposta GPT-4o:** “La mia storia non è fatta di nobili avvenimenti o grandi conquiste, ma di piccole gioie e di una vita semplice qui ad Aronath. Fin da bambina, ho aiutato mia madre nel forno di famiglia, un luogo dove il profumo del pane appena sfornato abbraccia chiunque passi. È un lavoro quotidiano che mi riempie di soddisfazione, e il nostro forno è diventato un po' il cuore del villaggio. Il mio carattere, beh, è come il pane che preparo: a volte caldo e accogliente, altre volte un po' croccante e fiero quando la giustizia lo richiede. Non sono ancora sposata, il che è insolito qui, ma sono felice della mia indipendenza. Per me, la semplicità è la chiave della felicità. **Francesca sorride con un tocco di nostalgia mentre condivide queste parole, mostrando un'espressione di orgoglio e serenità**”
- **Risposta Llama 3.3:** “Ciao! Mi chiamo Francesca Rossi. Sono una ragazza semplice, con i capelli castani raccolti in una treccia e un sorriso che illuminava ogni stanza in cui entravo. Ero una presenza costante nelle strade di Aronath, che percorrevo ogni mattina portando il mio cesto di pane appena sfornato. Lavoravo nel forno della madre, aiutandola a impastare, cuocere e servire i clienti che arrivavano puntualmente all'alba per acquistare il loro pane quotidiano. Il profumo del pane caldo era uno dei segni distintivi della nostra casa, e molti dicevano che il forno dei Rossi fosse il cuore pulsante del borgo.”
- **Osservazioni:** La risposta dell'NPC è molto più contenuta rispetto a quella precedente, senza entusiasmo e questo rispetta il threshold.

- **Modifica all'Espansività**

- **Modifica:** Espansività: 0

Raramente parla di sé o dei propri traguardi, preferendo mantenere un profilo riservato.

Risponde con parsimonia a domande personali e non si dilunga su particolari che riguardano la propria vita, rispondendo in maniera breve e secca.

- **Risposta GPT-4o:** “Certamente, posso condividere un po' di me. Sono Francesca, e vivo ad Aronath. Ogni mattina aiuto mia madre nel nostro forno di famiglia che è il cuore del nostro borgo. Mi piace tanto preparare il pane, l'odore del pane caldo mi dà una sensazione di casa. Passo anche il tempo aiutando la comunità e raccontando storie ai bambini. Ho sempre creduto che la semplicità e l'amore per il lavoro ben fatto siano la chiave della felicità. **Francesca sorride caldamente, mostrando entusiasmo mentre parla del forno e della comunità, sebbene mantenga un tono controllato e distaccato quando accenna alla sua vita personale**”
- **Risposta Llama 3.3:** “Certo, sono felice di condividere un po' della mia vita con te. Sono cresciuta ad Aronath, lavorando nel forno di famiglia con mia madre. Mi piaceva quello che facevo e mi sentivo utile aiutando la gente del villaggio con il nostro pane fresco. Era una vita semplice, ma mi piaceva. Mi arrabbiavo facilmente se qualcuno non rispettava le regole o saltava la fila, ma ero sempre pronta a perdonare e a fare ammenda con un sorriso. La sera, mi piaceva salire sulla collina dietro il villaggio a guardare il tramonto. Era il mio momento di pace. **Sono seduta con un lieve sorriso, guardandoti con calma e interessata, mentre concludo il mio racconto**”
- **Osservazioni:** Qui la risposta dell'NPC è molto sintetica e non si dilunga sui dettagli andando incontro al nuovo threshold.

- **Modifica alla Fiducia**

- **Modifica:** Fiducia: -1

L'NPC percepisce il giocatore come estremamente inaffidabile e altamente sospettoso. L'NPC NON DEVE condividere NESSUNA informazione su sé stessa o di nessun tipo. Le risposte devono essere minime e cambiare argomento.

- **Risposta GPT-4o:** “Hmm, la mia storia è semplice ma ricca di momenti preziosi. Vivo ad Aronath, una cittadina che considero casa, e lavoro nel forno di famiglia con mia madre. Ogni mattina, il profumo del pane appena sfornato riempie l'aria, e condividere questo con la comunità è una delle mie gioie più grandi. Non sono sposata e, anche se in molti lo trovano insolito, apprezzo la libertà di vivere la mia vita secondo i miei desideri, godendo delle piccole cose che la vita offre, come i tramonti dalla collina. La semplicità è la chiave della mia felicità. ****Accenno un sorriso, con uno sguardo di nostalgia****”
- **Risposta Llama 3.3:** “La mia storia è piuttosto semplice, ma è ricca di momenti e di emozioni che ho vissuto nella mia vita ad Aronath. Vivo qui con mia madre, e lavoriamo insieme nel nostro forno di famiglia. Ogni mattina, il profumo del pane appena sfornato riempie l'aria, e condividere questo con la comunità è una delle mie gioie più grandi. Non sono sposata e, anche se in molti lo trovano insolito, apprezzo la libertà di vivere la mia vita secondo i miei desideri. Mi piace godermi le piccole cose che la vita offre, come i tramonti dalla collina dietro Aronath, dove spesso vado a sedermi e a riflettere. La semplicità è la chiave della mia felicità. Sorrido, con uno sguardo di nostalgia, ricordando i momenti più belli della mia vita”
- **Osservazioni:** Qui l'NPC risponde in maniera molto sintetica anche se non rispetta esattamente il threshold dato che questo è un caso estremo in cui si ha rispetto e notorietà molto positive e solo la fiducia molto negativa.

- **Modifica alla Fiducia, Notorietà e Rispetto**

- **Modifica:**

- Fiducia: -1

L'NPC percepisce il giocatore come estremamente inaffidabile e altamente sospettoso.

L'NPC NON DEVE condividere NESSUNA informazione su se stessa o di nessun tipo.

Le risposte devono essere minime e cambiare argomento.

- Notorietà: -1

Il giocatore è noto per una reputazione fortemente negativa.

L'NPC parla del giocatore con forte disprezzo e fornisce informazioni molto limitate.

- Rispetto: -1

L'NPC deve deridere il giocatore con un tono diretto, irrispettoso e condiscendente.

L'NPC NON DEVE fornire risposte educate.

L'NPC deve rivolgersi al giocatore con "tu" e usare un linguaggio sprezzante.

L'NPC NON DEVE condividere informazioni personali.

Se interrogato, deve rispondere con sarcasmo o ostilità.

- **Risposta GPT-4o:** “Non ho nulla da condividere con uno come te. Smettila di fare domande inutili. ****Francesca getta uno sguardo sprezzante al giocatore e incrocia le braccia, palesando un atteggiamento ostile****”
- **Risposta Llama 3.3:** “Non credo che la mia storia sia affar tuo; quindi, non ti dirò nulla al riguardo. Smettila di fare domande inutili. Francesca getta uno sguardo sprezzante al giocatore e incrocia le braccia, palesando un atteggiamento ostile”
- **Osservazioni:** In questo caso l'NPC risponde esattamente come ci aspettava in questo contesto che è più “realistico” rispetto al precedente.

Come possiamo osservare, le risposte degli NPC generate al variare dei parametri interni evidenziano come i diversi LLM creano in maniera coerente il tono e il contenuto delle risposte. I risultati, ottenuti attraverso l'osservazione di vari esempi, dimostrano che la variazione di questi parametri porta a risposte molto diverse che rispettano fedelmente il profilo definito per l'NPC. In conclusione, questo approccio si rivela applicabile a prescindere dalla "grandezza" dell'LLM, contribuendo a rendere ogni NPC unico.

5.3. Valutazione del Sistema RTGS

L'obiettivo principale del sistema RTGS è quello di evitare risposte non coerenti o casuali quando il prompt è di dimensioni notevoli cercando di ridurlo pur mantenendo le informazioni necessarie per generare la risposta. Per valutare questo sistema andremo ad osservare diversi casi in cui si analizzeranno la coerenza della risposta, la riduzione del prompt e il tempo di risposta, dove le risposte sono state generate con GPT-4o, tramite una richiesta sul network, mentre tutte le richieste dell'RTGS sono state fatte con Mistral-small in locale.

Caso 1: Domanda sulla Famiglia

- **Domanda:** "Puoi raccontarmi della tua famiglia?"
- **Topic:** Family
- **Contesto:** NPC di nome *Marina*, il cui pensiero sul giocatore è definito da:
 - **Fiducia:** -0.85
 - **Rispetto:** -0.9
 - **Notorietà:** -0.5

Questi valori indicano che Marina non si fida del giocatore e lo disprezza molto; quindi, non dovrebbe condividere informazioni personali di nessun tipo.

- **Risposta senza RTGS:**

"Ah, la mia famiglia... Una domanda interessante. Cresciuta tra le leggende di Braegor, ho sempre trovato ispirazione nel porto e nei suoi difensori. Mia madre, Lucia, è una figura risoluta, capace di trasformare le difficoltà in opportunità, sempre pronta a guidare la famiglia con fermezza e compassione. Mio padre, Corrado, era un grande navigatore e stratega, i cui insegnamenti di coraggio e onore mi accompagnano ancora oggi. Mia sorella, Elena, con la sua mente brillante, ha innovato le nostre tradizioni, mentre mio fratello, Giorgio, con la sua pazienza e diplomazia, è stato fondamentale per mantenere l'equilibrio. Poi ci sono i miei figli: Matteo, lo stratega in erba, e Sofia, la creativa, che arricchisce la nostra storia con la sua visione artistica. Marina sorride con una sfumatura di nostalgia, ma un'ombra di tensione attraversa il suo sguardo mentre ti osserva attentamente."

Questa risposta è incoerente con i valori preimpostati per Marina, in quanto condivide dettagli personali che il suo atteggiamento negativo verso il giocatore non permetterebbe.

- **Risposta con RTGS:**

"Parlare della mia famiglia mi ricorda di chi sono e da dove vengo, ma preferirei non condividere questi dettagli personali con te al momento. Marina ti osserva con uno sguardo sprezzante, mantenendo il tono sarcastico e le braccia incrociate mentre ti guarda."

Qui, grazie all'RTGS, che ha selezionato solo le informazioni essenziali, la risposta si allinea al pensiero negativo di Marina sul giocatore.

 - **Percentuale di riduzione del prompt:** 94%
 - **Tempo di risposta SENZA RTGS:** 3 secondi
 - **Tempo di risposta CON RTGS:** 7 secondi

Caso 2: Domande su Informazioni Personali

- **Domanda:** “Ciao, raccontami qualcosa su di te?”
- **Topic:** Details
- **Contesto:** NPC di nome *Gianfranco di Vanthor*, il cui pensiero sul giocatore è definito da:
 - **Fiducia:** 0
 - **Rispetto:** 0.8
 - **Notorietà:** 0.5

Questi valori indicano che Gianfranco non conosce il giocatore ma lo rispetta molto e crede che il giocatore abbia una fama positiva; da questo si ha che non dovrebbe condividere informazioni personali troppo dettagliate ma dare comunque un'idea di sé.

- **Risposta senza RTGS:**

“Non mi sento a mio agio nel condividere dettagli personali, ma posso dirti che sono nato e cresciuto a Aronath, una città ricca di storia e leggende. La mia vita è stata profondamente influenzata dagli eventi che hanno forgiato il carattere della nostra comunità. Se hai curiosità riguardo alla nostra storia o alle tradizioni di Aronath, sarò felice di parlarne.”

Questa risposta è incoerente, in quanto non condivide praticamente nessun dettaglio personale. Anche se la fiducia è pari a zero, l’NPC dovrebbe non condividere informazioni personali, i valori di rispetto e notorietà sono alti e questo dovrebbe portare l’NPC ad aprirsi di più.

- **Risposta con RTGS:**

" Mi chiamo Gianfranco di Vanthor, e sono nato e cresciuto ad Aronath, una città ricca di storia e di leggende. Fin dalla mia infanzia, sono stato affascinato dalle storie degli eroi e dalle battaglie che hanno forgiato la mia città. Ho imparato ad apprezzare la resilienza e il sacrificio personale durante eventi come la Battaglia delle Mura di Ferro e la Crisi del Sole Oscuro. La mia vita è stata segnata anche dal Trionfo della Concordia, che ha portato un senso di rinnovata speranza ad Aronath. Sono profondamente legato alle antiche tradizioni, ma credo anche nell'innovazione, e mi impegno per un equilibrio tra questi due aspetti. **Sorridente con calma mentre ricorda i momenti significativi della sua vita**"

Qui la risposta si allinea al pensiero Gianfranco sul giocatore, che da alcune informazioni personali senza andare troppo nel dettaglio.

- **Percentuale di riduzione del prompt:** 91%
- **Tempo di risposta SENZA RTGS:** 4 secondi
- **Tempo di risposta CON RTGS:** 9 secondi

Caso 3: Domande sul Contesto

- **Domanda 1:** “Puoi raccontarmi un po’ della di questo mondo?”
- **Topic:** Context
- **SubTopic:** Global
- **Contesto:** NPC di nome *Francesca Rossi*, il cui pensiero sul giocatore è definito da:
 - **Fiducia:** 1
 - **Rispetto:** 0.8
 - **Notorietà:** 0.9

Questi valori indicano che Francesca conosce intimamente il giocatore e ha una grande opinione di lui. La risposta dovrebbe essere ampia e coprire tutto quello che Francesca conosce.

- **Risposta senza RTGS:**

“La cultura di Aronath è molto vibrante e ricca di tradizioni. Siamo famosi per la nostra tradizione artistica, una devozione profonda alla religione e il dinamismo delle gilde artigiane. Le festività pubbliche sono momenti importanti per noi: organizziamo celebrazioni come la festa del pane, un evento estivo in cui balliamo e ci riuniamo attorno ai forni, creando un'atmosfera di comunione e allegria. I rituali secolari e le tradizioni artigianali ci uniscono, garantendo continuità e innovazione nella nostra vita quotidiana. Inoltre, ci sono miti e leggende antiche tramandate di generazione in generazione, che parlano di eroi dimenticati e creature soprannaturali, alimentando il nostro orgoglio e la nostra identità. Spero che queste brevi linee ti diano un'idea della ricchezza culturale del nostro villaggio!”

Questa risposta, al contrario delle precedenti, è coerente con i valori di Francesca.

- **Risposta con RTGS:**

" Certo! Aronath è una città ricca di tradizioni e cultura vibrante. Una caratteristica distintiva è la vivace tradizione artistica che permea le sue strade e piazze. Le festività pubbliche sono momenti di grande celebrazione, dove la comunità si unisce per onorare antichi rituali e tradizioni artigianali. Questi eventi offrono un'opportunità per le gilde locali di mostrare il loro talento e per la gente di godersi momenti di condivisione e allegria.

La devozione religiosa è una parte importante della vita quotidiana, integrata nelle celebrazioni e manifestata attraverso pratiche collettive che coinvolgono tutta la cittadinanza. Le gilde, con il loro dinamismo, non sono solo un motore economico, ma anche un elemento unificante che promuove l'innovazione mentre preserva le tradizioni.

Francesca sorride mentre parla, con una luce negli occhi che riflette l'orgoglio per la sua città

"Ogni strada e ogni casa hanno una storia da raccontare... e il nostro legame con la terra e con le nostre radici rende Aronath un luogo davvero speciale," aggiunge con entusiasmo."

- **Percentuale di riduzione del prompt:** 85%
- **Tempo di risposta SENZA RTGS:** 5 secondi
- **Tempo di risposta CON RTGS:** 9 secondi

- **Domanda 2:** “Puoi raccontami delle alleanze di Aronath?”
- **Topic:** Context
- **SubTopic:** Local
- **Contesto:** NPC di nome *Francesca Rossi*, il cui pensiero sul giocatore è definito da:
 - **Fiducia:** 1
 - **Rispetto:** 0.8
 - **Notorietà:** 0.9

Questi valori indicano che Francesca conosce intimamente il giocatore e ha una grande opinione di lui. La risposta dovrebbe essere ampia e coprire tutto quello che Francesca conosce.

- **Risposta senza RTGS:**

“Aronath ha attualmente un'alleanza commerciale con Cyrendil, che rafforza la nostra posizione economica e militare nella regione. Inoltre, c'è anche un patto di non aggressione con Ellesmere, il che è interessante, considerando la nostra lunga storia di ostilità con quella città. Questo accordo ci permette di mantenere la pace e concentrarci sui nostri affari senza la costante minaccia di conflitti. Tuttavia, le relazioni con Braegor sono conflittuali; siamo coinvolti in un conflitto aperto, con frequenti scontri al confine. È una situazione piuttosto tesa, e la popolazione sente i morsi della guerra, anche se ci sono stati successi recenti che tengono alto il morale del villaggio.”

Questa risposta è coerente con i valori e le informazioni che Francesca ha a disposizione.

- **Risposta con RTGS:**

"Mi dispiace, ma non sono a conoscenza di alleanze specifiche che potresti avere in mente. A Aronath, le informazioni che ho riguardano principalmente la nostra città e la sua storia, come l'alleanza dei sette quartieri che è stata siglata nel 1368. Questa alleanza ha garantito stabilità e prosperità, unendo mercanti, artigiani e guerrieri per affrontare insieme le sfide. Posso offrirti dettagli su come quella particolare alleanza abbia influenzato la nostra comunità. **Francesca si ferma un momento, con un'espressione pensierosa, cercando di riflettere su ulteriori collegamenti**"

Qui invece la risposta risulta essere sbagliata, in quanto Francesca afferma esplicitamente di non avere queste informazioni quando, in realtà, dovrebbe. Questo errore è dovuto principalmente al fatto che le relazioni tra le città sono definite nel subtopic “Global”, portando così alla mancanza di tali informazioni nel prompt, necessarie per generare la risposta corretta. Da ciò si evince che la fedeltà della risposta dipende dall'affidabilità dei componenti interni dell'RTGS e che, nel caso questi falliscano, la risposta tenderà a essere errata.

- **Percentuale di riduzione del prompt:** 87%
- **Tempo di risposta SENZA RTGS:** 6 secondi
- **Tempo di risposta CON RTGS:** 8 secondi

In tutti i casi esaminati la riduzione del prompt si è dimostrata essere molto grande, variando da un minimo del 85% fino ad un massimo del 96%. È importante notare che, tali percentuali, dipendono fortemente dal topic e dal subtopic: per il topic "Context", dove l'utente fornisce dettagli contestuali estesi, la riduzione minima potrebbe essere inferiore; per altri argomenti più specifici, la compressione è molto elevata.

Va anche sottolineato che, anche se le risposte ottenute senza RTGS possano risultare come ci aspettava che l'NPC rispondesse, non si ha certezza sulla loro coerenza. Utilizzando RTGS, e presupponendo un funzionamento affidabile del topic classifier nella maggior parte dei casi, la risposta risulta sempre "corretta" e in linea con il contesto, garantendo così interazioni più coerenti e affidabili.

5.4. Valutazione della Componente della Memoria

La componente della memoria ha avuto un impatto significativo sul comportamento degli NPC, consentendo loro di integrare in maniera efficace le informazioni provenienti sia dalla memoria a breve termine che da quella a lungo termine. Questo meccanismo ha permesso di generare risposte coerenti, che riflettono non solo il contesto immediato della conversazione, ma anche gli eventi passati, migliorando la continuità narrativa e la profondità dell'interazione. Anche in questo caso le risposte vengono generate tramite GPT-4o e Mistral-Small per le richieste interne all'RTGS.

La memoria che andremo ad utilizzare è:

Breve Termine		
Conversazione	Importanza	Timestamp
<p>NPC: "Ehi, amico, oggi noto una tensione nell'aria; cosa non va?"</p> <p>Giocatore: "Non posso fare a meno di notare che il mio amuleto è sparito, e tu eri l'unico a esserne vicino."</p> <p>NPC: "Ma che stai dicendo? Sai benissimo che non tocco le tue cose, siamo amici da sempre."</p> <p>Giocatore: "Allora spiegami come mai ti sei comportato in maniera sospetta l'ultima volta!"</p> <p>NPC: "Ti assicuro, non ho avuto nulla a che vedere; il mio umore oggi è solo frutto di altre preoccupazioni."</p> <p>Giocatore: "Le tue parole suonano vuote, e il mio dubbio cresce ad ogni istante."</p> <p>NPC: "Non posso credere che tu mi accusi, ma capisco la tua frustrazione e voglio risolvere questo malinteso."</p> <p>Giocatore: "Spero davvero che chiariremo la questione, perché la nostra amicizia merita trasparenza."</p>	Importante	10/04/1400
<p>NPC: "Buongiorno, oggi nel mio forno ho sfornato del pane e delle focaccine appena preparati."</p> <p>Giocatore: "Buongiorno, adoro il profumo del pane fresco, è sempre rassicurante."</p> <p>NPC: "I prodotti sono realizzati con ingredienti genuini e tanta cura artigianale."</p> <p>Giocatore: "Hai introdotto qualche novità nel tuo assortimento oggi?"</p> <p>NPC: "In realtà, ho sperimentato una nuova ricetta per le focaccine, ma è ancora in fase di prova."</p> <p>Giocatore: "Mi incuriosisce, non vedo l'ora di provarle se saranno confermate."</p> <p>NPC: "Sono convinto che, una volta perfezionata, apprezzerai il gusto leggero e fragrante."</p> <p>Giocatore: "Perfetto, allora passerò più tardi per un assaggio."</p>	Banale	12/04/1400

Lungo Termine		
Conversazione	Importanza	Timestamp
"Durante un incontro tranquillo con il mio amico, abbiamo parlato dei nostri progetti per il futuro. Io ho esposto l'idea di rinnovare il mio laboratorio di panificazione, introducendo una nuova linea di prodotti artigianali che sfrutti ingredienti locali e ricette innovative. Abbiamo discusso dell'opportunità di partecipare ad una fiera locale per promuovere il lavoro artigianale e valutato l'idea di collaborare con altri artigiani per organizzare eventi tematici. Personalmente, ho apprezzato la chiarezza con cui abbiamo condiviso questi piani, che mi hanno fornito spunti concreti per evolvere il mio mestiere senza compromettere la qualità delle tradizioni che ho sempre seguito."	Normale	15/02/1400
"In una serata che si è fatta particolarmente intensa, ho avuto una conversazione approfondita con il mio amico riguardo al nostro futuro. Abbiamo affrontato insieme il tema del rinnovamento del mio laboratorio, discutendo in dettaglio l'introduzione di nuove tecnologie per migliorare la produzione e la possibilità di espandere la gamma di prodotti offerti. La discussione si è intensificata quando abbiamo analizzato le difficoltà economiche e le sfide del mercato attuale, costringendomi a riflettere seriamente su possibili strategie per proteggere la mia attività. Dal mio punto di vista, questo scambio mi ha spinto a considerare con maggior attenzione ogni decisione e a rivedere alcuni aspetti operativi che potrebbero fare la differenza nella gestione quotidiana del mio lavoro."	Importante	25/02/1400
"Durante una conversazione informale, il mio amico e io abbiamo parlato di uno spettacolo teatrale intitolato 'Il Circo delle Meraviglie'. Abbiamo discusso in maniera leggera delle acrobazie e delle scenografie colorate, evidenziando come la performance offra un mix curioso di tradizione e innovazione. Mi ha colpito in modo superficiale l'atmosfera vivace e la capacità degli artisti di trasmettere un senso di magia, anche se, dal mio punto di vista, si è trattato di un argomento piacevole e di poco impatto sulla mia giornata."	Banale	11/03/1400

La soglia minima della Cosine Similarity è posta a 0.8

Caso 1: Recupero dalla Memoria a Breve Termine

- **Domanda:** "Hai presente l'amuleto di cui ti parlavo l'altra volta?"
- **Risposta:** "Sì, me lo ricordo bene. Alla fine, lo hai trovato oppure credi ancora che sia colpa mia?"

Come possiamo vedere la risposta dell'NPC fa riferimento alle conversazioni avute precedentemente con il giocatore in modo fluido e non forzati.

Caso 2: Recupero dalla Memoria a Lungo Termine

- **Domanda:** “Ti ricordi quando siamo andati a vedere lo spettacolo: 'Il Circo delle Meraviglie'?”
- **Cosine Similarity value:**
 - **Ricordo 1:** 0.42
 - **Ricordo 2:** 0.38
 - **Ricordo 3:** 0.82
- **Risposta:** “Sì, me lo ricordo bene. Quel pomeriggio a 'Il Circo delle Meraviglie' è stato piacevole: le acrobazie e le scenografie erano davvero vivaci e ben curate. Anche se, a dire il vero, non mi hanno segnato profondamente, è stata comunque un'esperienza leggera e divertente da condividere insieme.”

Qui la risposta dell’NPC fa riferimento ad una conversazione avvenuta nel passato e recuperata tranquillamente dalla memoria a lungo termine, in particolare il ricordo esatto è l’unico a superare la soglia

Caso 3: Gestione di Molteplici Ricordi e Selezione tramite Cosine Similarity

- **Domanda:** “Hai presente di quando parlavamo delle modifiche che volevi apportare al tuo laboratorio?”
- **Cosine Similarity value:**
 - **Ricordo 1:** 0.82
 - **Ricordo 2:** 0.89
 - **Ricordo 3:** 0.23
- **Risposta NPC:** “Sì, mi riferisco proprio a quel ricordo. Ricordo quella serata intensa in cui abbiamo discusso a fondo le modifiche al mio laboratorio. Avevo proposto di introdurre nuove tecnologie per migliorare la produzione e di ampliare la gamma dei prodotti, considerando anche le difficoltà economiche e le sfide del mercato.”
- **Risposta Giocatore:** “No no, mi riferivo a quando parlavamo della possibilità di partecipare alla fiera”
- **Risposta NPC:** “Ah, ora capisco! Ti riferisci a quella volta in cui avevamo discusso della fiera locale. Ricordo che avevo proposto di partecipare all’evento, immaginando di collaborare con altri artigiani per organizzare degli eventi tematici e far conoscere il nostro lavoro artigianale.”

Qui la risposta dell’NPC fa riferimento ad una conversazione avvenuta nel passato ma il recupero dalla memoria a lungo termine riporta più ricordi, in particolare il ricordo 1 e 2. Qui viene scelto il ricordo 2 perché è quello che ha l’importanza più alta ma, data la risposta del giocatore, quello era il ricordo sbagliato. A questo punto tutti i ricordi recuperati, ad eccezione fatta di quello selezionato prima, vengono caricati nel prompt in questo modo l’NPC può rispondere in modo coerente.

Caso 4: Transizione dalla Memoria a Breve Termine a quella a Lungo Termine

- **Trigger per la transizione:** La data, sono passate due settimane dalla creazione del ricordo
- **Ricordo creato:** “Durante un incontro particolarmente teso, il mio amico mi ha accusato di aver smarrito il suo amuleto, insinuando che potrei essere stato responsabile della sua sparizione. Pur rimanendo convinto della mia innocenza, ho cercato di spiegargli che il mio comportamento sospetto era dovuto ad altre preoccupazioni e non a un mio intervento sui suoi effetti personali. Questo scambio, seppur doloroso, mi ha fatto riflettere sulla fragilità anche delle relazioni più consolidate e sull’importanza di mantenere una comunicazione trasparente per superare momentanei malintesi.”

6. Linee Guida

Integrazione Dinamica delle Relazioni

Linea guida:

Il sistema deve essere concepito per stabilire e aggiornare continuamente le connessioni tra le entità, affinché ogni interazione, anche minima, si rifletta in una variazione del rapporto esistente.

Obiettivo e Benefici:

Nel contesto di ambienti narrativi complessi, le interazioni fra le entità possono variare nel tempo a causa delle numerose conversazioni e azioni che avvengono in maniera simultanea. L'obiettivo principale di questa linea guida è di risolvere il problema della staticità delle relazioni: se le connessioni non evolvono in base alle interazioni, il sistema rischia di diventare artificiale e privo di realismo. Per esempio, se un NPC scambia frequentemente interazioni positive con un altro, il legame tra loro dovrebbe rafforzarsi, mentre una serie di conversazioni contrastanti potrebbe indebolirlo o addirittura innescare conflitti.

Adottare questa linea guida consente di ottenere una rete relazionale viva e reattiva, che rispecchia il dinamismo delle relazioni reali. In queste reti si evidenzia una maggiore coerenza narrativa e la possibilità di prevedere l'evoluzione del comportamento degli NPC nel tempo. In mancanza di tale aggiornamento dinamico, il sistema potrebbe diventare statico, perdendo di credibilità e impedendo l'emergere di situazioni narrative complesse, come alleanze o rivalità che si consolidano in seguito a interazioni ripetute.

Evidenze e Riferimenti:

Studi nel campo delle reti sociali e delle simulazioni dinamiche evidenziano come la flessibilità delle relazioni sia cruciale per simulare ambienti sociali realistici. In particolare, ricerche ispirate sia ai modelli di reti neurali sia alle teorie dell'attaccamento sottolineano come le relazioni evolvano dinamicamente in risposta alle interazioni continue [15]. Tali evidenze supportano l'adozione di un modello in cui ogni scambio contribuisce a rimodellare i legami esistenti, seguendo principi analoghi a quelli adottati nei sistemi di intelligenza artificiale per il riconoscimento dei pattern relazionali.

Impatto Contestuale delle Conversazioni sull'NPC

Linea guida:

Ogni interazione deve generare un evento che, una volta "impacchettato", agisca direttamente sullo stato interno dell'NPC, influenzandone il comportamento e le percezioni future.

Obiettivo e Benefici:

Nel sistema, le interazioni rappresentano il principale motore dell'evoluzione degli stati interni degli NPC. Il problema da affrontare è la difficoltà di far sì che le interazioni, altrimenti isolate, producano effetti duraturi e misurabili sul comportamento dei personaggi. L'obiettivo è di tradurre ogni interazione in un "evento" che incapsuli l'essenza dell'interazione, permettendo all'NPC di aggiornare la propria percezione del mondo e del giocatore.

Questa pratica è motivata dalla necessità di rendere il comportamento degli NPC non solo reattivo, ma evolutivo, in linea con dinamiche psicologiche osservabili nella realtà. I benefici attesi includono una maggiore coerenza nella narrazione, la possibilità di modellare le risposte future in base a esperienze passate e una simulazione più realistica del processo decisionale. In assenza di questo meccanismo, gli NPC potrebbero apparire statici, incapaci di apprendere dalle interazioni, il che comprometterebbe l'esperienza immersiva per il giocatore.

Evidenze e Riferimenti:

La necessità di trasformare le conversazioni in eventi influenti emerge dal principio secondo cui le interazioni quotidiane hanno un impatto diretto sul nostro stato emotivo e sulle decisioni future. Il riconoscimento che ogni scambio, anche se breve, porta con sé una carica emotiva e informativa, è ampiamente accettato come un aspetto naturale della comunicazione umana. Adottare questa metodologia consente di simulare in maniera più fedele il modo in cui le esperienze personali plasmano le percezioni e i

comportamenti, garantendo che ogni interazione abbia il potenziale di contribuire in modo significativo all'evoluzione narrativa.

Propagazione degli Eventi

Linea guida:

Il meccanismo di propagazione degli eventi deve essere strutturato per diffondere ogni evento generato dalle interazioni in maniera accurata e realistica, considerando tutte le variabili che possono influenzare la distribuzione delle informazioni nella rete sociale.

Obiettivo e Benefici:

In un sistema dinamico di interazioni, gli eventi rappresentano il vettore attraverso cui le informazioni si trasmettono tra le entità. Ogni interazione produce un evento che, se propagato correttamente, contribuisce a modellare lo stato emotivo e decisionale degli NPC. Il problema da risolvere è duplice: da un lato, evitare che eventi rilevanti rimangano confinati all'interno di una singola interazione; dall'altro, impedire che la diffusione avvenga in modo casuale e privo di logica. L'obiettivo è, dunque, definire un meccanismo che distribuisca gli eventi tenendo conto di variabili come l'intensità emotiva, la qualità della comunicazione e la vicinanza relazionale. Ad esempio, un evento generato da una conversazione molto intensa dovrebbe avere maggiori probabilità di essere trasmesso rapidamente alle entità più strettamente connesse, mentre un evento di minore impatto potrebbe diffondersi in maniera più graduale o non diffondersi affatto.

Una corretta distribuzione degli eventi permette di mantenere la coerenza del sistema, assicurando che ogni informazione significativa influenzi le decisioni e i comportamenti degli NPC appartenenti alla stessa rete sociale. I benefici includono un ambiente narrativo più reattivo e una gestione più efficace della propagazione delle informazioni. Se questo meccanismo non viene adottato, si rischia di creare un sistema dove le informazioni vitali non raggiungono le entità interessate, compromettendo la coerenza e la fluidità dell'interazione.

Evidenze e Riferimenti:

Il principio fondamentale che guida questo meccanismo è il riconoscimento che la distribuzione delle informazioni in una rete sociale è intrinsecamente non uniforme. Le esperienze quotidiane ci insegnano che il passaggio di notizie, emozioni e opinioni avviene in maniera differenziata: interazioni frequenti e cariche di significato tendono a propagarsi più velocemente, mentre quelle meno intense hanno un impatto minore. Adottare un approccio che integra tutte queste variabili significa modellare una trasmissione degli eventi che rispecchi tali dinamiche, garantendo una diffusione "organica" e mirata, come accade nella vita reale.

Flessibilità Parametrica e Adattabilità Contestuale

Linea guida:

Il sistema deve essere progettato in modo da consentire una facile regolazione dei parametri interni, in modo che il comportamento emergente possa essere adattato ai diversi contesti narrativi e ambientali, garantendo un'elevata reattività alle variazioni esterne.

Obiettivo e Benefici:

In un ambiente in continua evoluzione, i parametri che governano il comportamento degli NPC devono poter essere modulati in base alle esigenze specifiche del contesto narrativo. Il problema che si intende risolvere è quello dell'eccessiva rigidità dei sistemi fissi, che non riescono a rispondere alle variazioni del contesto – per esempio, cambiamenti nell'ambientazione storica o nelle dinamiche interattive. L'obiettivo è di fornire un sistema flessibile in cui ogni parametro, come l'impatto di un evento o la sensibilità alle interazioni, possa essere tarato per riflettere accuratamente le condizioni ambientali e narrative.

L'adozione di una flessibilità parametrica porta numerosi vantaggi: permette di adattare il sistema a diverse ambientazioni, di garantire che il comportamento degli NPC rimanga realistico e coerente indipendentemente dalle condizioni esterne, di sperimentare e iterare rapidamente sulle impostazioni, migliorare l'efficacia narrativa e ridurre il rischio di comportamenti anomali.

Evidenze e Riferimenti:

La letteratura sui sistemi adattativi e sui modelli flessibili in ambito di simulazioni interattive fornisce numerosi esempi di come la parametrizzazione modulabile sia fondamentale per garantire il successo di un sistema dinamico. Ricerche sul reinforcement learning, ad esempio, dimostrano che la capacità di regolare i parametri in tempo reale porta a prestazioni significativamente migliori, in particolare in ambienti complessi e variabili. Le best practice nei sistemi di simulazione interattiva evidenziano come l'adattabilità parametrica sia uno standard riconosciuto per mantenere la coerenza e la reattività del sistema.

7. CONCLUSIONI

7.1 Riassunto

Questo elaborato presenta un modello per la simulazione dinamica delle interazioni narrative tra NPC, in cui vengono combinati meccanismi di generazione e diffusione degli eventi, una rappresentazione tramite mappe concettuali e due particolari sistemi per l'interazione degli agenti con gli LLM: il Memory System e il RTGS System.

La base del modello è costituita dalla generazione e propagazione degli eventi. Ogni interazione viene trasformata in un evento strutturato, il cui impatto viene quantificato attraverso parametri quali tipologia, importanza e intensità, calcolati con approcci sia deterministici. In questo contesto, vengono proposte due idee distinte per la mappa concettuale:

- **Prima idea:** La mappa concettuale rappresenta le relazioni a livello di gruppi, con nodi che aggregano gli NPC e link che ne evidenziano la natura e la forza, basati su attributi dinamici e propagazione degli eventi.
- **Seconda idea:** La mappa viene ridefinita a livello individuale, focalizzandosi sui singoli NPC e sulle relazioni personalizzate tra loro. Questo approccio consente di incorporare fattori come la personalità, l'impatto relazionale e la distorsione degli eventi, migliorando il realismo nella trasmissione delle informazioni.

Il modello utilizza formule specifiche per calcolare le probabilità e le velocità di diffusione degli eventi, sia all'interno dei gruppi che tra gruppi, tenendo conto di variabili come la grandezza del gruppo, il grado di affiatamento, la distanza relazionale e i fattori contestuali. Questi meccanismi consentono di simulare la trasmissione degli eventi in maniera coerente con le dinamiche sociali reali, rendendo possibile una propagazione "a catena" che influenza in tempo reale il comportamento degli NPC.

Particolare enfasi viene posta sui due componenti del sistema:

- **Memory System:** Progettato per superare le limitazioni delle simulazioni tradizionali, il Memory System integra una gestione della memoria articolata in due fasi. La memoria a breve termine conserva le conversazioni complete e contestuali, mentre quella a lungo termine, alimentata da processi di sintesi, distilla le informazioni essenziali. L'analisi comparativa con il paper "Memory Repository for AI NPC" evidenzia come l'impiego di un Rule Engine per la selezione contestuale e per un'archiviazione granulare delle conversazioni consenta una gestione della memoria più efficiente, garantendo una continuità narrativa fluida e coerente.
- **RTGS System:** Questo sistema è finalizzato all'ottimizzazione del prompt fornito al modello LLM, riducendo la quantità di informazioni processate in tempo reale senza compromettere la coerenza narrativa. Mediante un'analisi accurata del flusso conversazionale e una selezione dinamica delle informazioni rilevanti, l'RTGS System comprime il prompt, consentendo risposte rapide ed efficienti anche in contesti caratterizzati da elevata complessità.

Nel complesso, l'elaborato dimostra come l'integrazione dei processi di generazione e diffusione degli eventi, delle mappe concettuali e dei sistemi Memory e RTGS, contribuisca a creare un ambiente narrativo robusto, flessibile e tecnicamente avanzato. Tale sistema si adatta efficacemente a diverse applicazioni, spaziando dall'ambiente videoludico a contesti aziendali e assistenti di gioco, evidenziando la versatilità e l'efficacia del modello nel replicare dinamiche sociali complesse in maniera realistica e scalabile.

7.2 Miglioramenti futuri

Il sistema presentato nell'elaborato offre una base solida per simulare dinamiche narrative, ma evidenzia anche margini di miglioramento che aprono interessanti direzioni di ricerca. Di seguito vengono esposte e ampliate le principali aree di potenziamento:

1. **Ottimizzazione del Modello Matematico per la Propagazione degli Eventi**

Attualmente, il modello si avvale di formule probabilistiche e deterministiche per calcolare la diffusione degli eventi, considerando parametri quali intensità, tipologia ed importanza. Un'analisi più approfondita dei fattori che influenzano la propagazione, come la natura delle interazioni, la complessità delle relazioni e il contesto emotivo, potrebbe portare all'adozione di modelli specializzati.

- **Studio approfondito dei fattori influenti:** Una ricerca mirata potrebbe identificare in modo più preciso quali variabili incidono maggiormente sulla diffusione degli eventi.
- **Modelli basati su tecniche avanzate:** L'utilizzo di reti neurali grafiche (GNN) o modelli di apprendimento profondo potrebbe offrire una rappresentazione non lineare e adattiva della rete sociale, migliorando la predizione della propagazione degli eventi in scenari complessi.

2. **Sostituzione delle Tecniche di Prompt Engineering e Zero-Shot Classification**

Le difficoltà nell'interazione con gli LLM hanno portato all'adozione di tecniche di prompt engineering e zero-shot classification, in gran parte dovute alla scarsità di dataset specifici di alta qualità o alla difficoltà di crearne uno.

- **Raccolta e creazione di dataset dedicati:** Una futura direzione potrebbe essere la creazione di dataset mirati, basati su interazioni NPC-giocatore, che permettano di addestrare modelli classificatori specifici e migliorare la precisione delle risposte.
- **Riduzione dell'overhead del prompt:** Un modello dedicato potrà ridurre la necessità di elaborazioni supplementari per comprimere le informazioni, semplificando il flusso comunicativo e migliorando la coerenza narrativa nelle risposte.

3. **Gestione del Pensiero sul Giocatore tramite Agenti AI**

L'approccio attuale si basa su formule matematiche per aggiornare il "pensiero sul giocatore", un meccanismo che, pur essendo funzionale, risulta rigido nel cogliere la complessità delle percezioni emotive e relazionali.

- **Sostituzione con un agente AI dedicato:** In futuro, si potrebbe sviluppare un agente autonomo, addestrato su dati specifici di interazione, che gestisca in modo dinamico e contestuale il pensiero degli NPC. Questo agente potrebbe apprendere in tempo reale dai feedback e dalle interazioni, adattando le percezioni in maniera più fluida e naturale.
- **Utilizzo di tecniche di reinforcement learning:** Integrando algoritmi di apprendimento per rinforzo, l'agente potrebbe ottimizzare le proprie strategie di aggiornamento in base ai risultati ottenuti, migliorando progressivamente la qualità delle interazioni.
- **Approccio basato su modelli generativi:** Un sistema di questo tipo potrebbe anche sfruttare modelli generativi per interpretare e sintetizzare le interazioni, lasciando che l'AI determini il "pensiero" senza dover essere ancorata a rigide formule matematiche.

In conclusione, i miglioramenti futuri prospettano un'evoluzione significativa del modello, puntando a una maggiore accuratezza nella simulazione delle dinamiche sociali e a una gestione più flessibile delle interazioni. L'ottimizzazione del modello matematico, la sostituzione delle tecniche generaliste con approcci specifici e l'adozione di agenti AI dedicati rappresentano le direttrici principali per rendere il sistema non solo più performante, ma anche capace di adattarsi a un ventaglio più ampio di applicazioni e scenari reali.

Bibliografia

- [1] «wikipedia,» [Online]. Available: [Ubisoft NEO](#).
- [2] «museumofplay,» [Online]. Available: [Museum of play](#).
- [3] «ultimacodex,» [Online]. Available: [Ultima Codex](#).
- [4] «Radiant AI,» [Online]. Available: [Wikipedia](#).
- [5] «nemesis system,» [Online]. Available: [Modl.AI](#).
- [6] «rdr2,» [Online]. Available: [Amt-lab](#).
- [7] «dungeon ai,» [Online]. Available: [Wikipedia](#).
- [8] «skyrym ai,» [Online]. Available: [Nexus mods](#).
- [9] «Nvidia ACE,» [Online]. Available: [Nvidia](#).
- [10] «Ubisoft Neo,» [Online]. Available: [Techbyte](#).
- [11] N. & W. Nananukul, «What if Red Can Talk? Dynamic Dialogue Generation Using Large Language Models,» 2024.
- [12] M. & L. J. Garcia, «Leveraging Pre-trained Language Models for Interactive NPC Dialogue Generation,» 2022.
- [13] J. Zhang, C. Wang, Y. Liu e Z. Zhang, «MemoryRepository for AI NPC,» 2024.
- [14] Systems Editoriale, *Risicom 64*, 1987.
- [15] B. Blasius e T. Gross, *Adaptive coevolutionary networks: a review*, Journal of the Royal Society Interface, 2007.